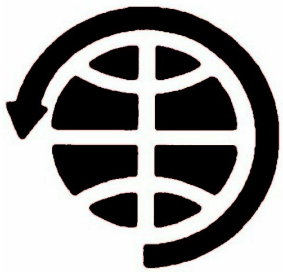


AMSAT®

RADIO AMATEUR SATELLITE CORPORATION

# PROCEEDINGS OF THE

# AMSAT-NA

## 35th Space Symposium and AMSAT-NA Annual Meeting

**October 27-29**

# 2017

**Reno, Nevada**

Copyright 2017
The Radio Amateur Satellite Corporation

All Rights Reserved. No part of this work may be reproduced
in any form except by written permission of the publisher.
All rights of translation are reserved.

Printed in USA

First Edition

## TABLE OF CONTENTS

Welcome - Barry Baines WD4ASW..... i

Silent Keys..... iii

**Section A: AMSAT Satellite Projects and Technology**

Ascent Ground and Satellite Demonstration
Ray Roberge, WA1CYB, Howie DeFelice, AB2S ….....1

Frequency Calibration for SDRs - Without GPS
Ray Roberge, WA1CYB..... 9

A Fault-Tolerant, Redundant IHU with a Frequency Agile Digital Command Transceiver
Zach Metzinger, N0ZGO, Jonathan Brandenburg, KF5IDY, Bill Reed, NX5R,
Jordan Trewitt, KF5COQ, Jesse Marroquin, K5JXM..... 13

Evolution of the VITA 74 Standard (VNX) for “SpaceVNX” CubeSAT Applications
Bill Ripley, KY5Q, Patrick Collier..... 47

Fox-1E (RadFxSat-2) Telemetry
Burns Fisher, W2BFJ..... 61

Techniques for Interpreting Fox-1 MEMS Gyro Readings Or Experimenting with an Experiment
Burns Fisher, W2BFJ..... 67

Optimally Calculating and Plotting Satellite Subpoints for AO-85 (Fox-1A) CPU Resets
Douglas D. Quagliana, KA2UPW/5..... 75

A Whole Orbit Data (WOD) Simulation based on Satellite Orbit Prediction Software
Carl E. Wick, N3MIM..... 85

An Update on Past and Future Projects from AMSAT-UK
David Bowman, G0MRF..... 93

Introduction to SatNOGS
Corey Shields, KB9JHU..... 99

Amateur and Weather Satellite Ground Station Development Activities at Holy Angel University
L. M. Almazan, WA6LOS, A. I. Parungao, DV3UZT, J. P. de Guzman, DW3CQK,
L. P. Paz, DW3CQF, R. P. Gopez, DW3CQC, K. S. Gueco, DW3CQI,
R. J. Sarmiento, DW3CQB, R. M. Pascual, DW3CQJ, J. A. Arciga, Cali Smile,
J. Y. Castaneda, H. P. Yalung.....103

Albuquerque Public Schools STEM Trajectory Initiative CubeSAT Project
Alma Ripley.....121

AMSAT NxtGen Crystal Radio
Joe Spier, K6WAO..... 129

**Section B: AMSAT Fox-1 Systems Engineering Documentation** –
**by the AMSAT FOX Engineering Team**

AMSAT Engineering Documentation Update
Jerry Buxton NØJY..... 157

Fox 1E (RadFxSat-2) System Requirements Specification..... 159

Fox 1E Linear Transponder Communications Performance Analysis..... 169

Fox 1E Downlink Specifications..... 189

RadFxSat-2 IHU to Experiment 1 Interface Control Document..... 195

RadFxSat-2 / Fox IHU Interface Control Document..... 211

BPSK Modulator I&Q Prototype 2..... 223

Fox 1E Transponder Schematic v1.1..... 225

Fox 1 Mechanical Drawings..... 229

AMSAT®
The Radio Amateur Satellite Corporation
10605 Concord Street, #304
Kensington, MD 20895
Phone 301-822-4376 • Fax 301-822-4371
Toll Free 1-888-322-6728

Welcome to Reno!

It is with great pleasure that I personally welcome you to the 2017 AMSAT Space Symposium and Annual Meeting. This is the 35th Symposium, reflecting a long tradition (started in the early 1980's) to provide an annual forum for amateur satellite enthusiasts to gather and share their ideas. AMSAT is proud of the technical content of the papers presented over the years reflecting the inventiveness of our volunteers and the significant contribution of presenters to "advancing the radio art."

Holding our Symposium at Reno this year is certainly fitting given that Reno and the Sierra Nevada are known for the frontier that was created by intrepid individuals who forged new paths and created new opportunities. Likewise, AMSAT's pioneering innovation in small satellite design and applications for amateur radio and science is also well recognized. We expect four of our Fox-1 class satellites to be in orbit within the next eight months, each with their own personality and orbital characteristics. RadFXSat-1/Fox-1B is currently scheduled for launch November 10 from Vandenberg AFB, CA as part of ELaNa-XIV. Fox-1Cliff is a commercial launch on the Spaceflight SSO-A mission from Vandenberg expected Second Quarter 2018. Fox-1D is also a commercial launch arranged by Spaceflight, Inc. flying on a PSLV (Polar Satellite Launch Vehicle) launch late this year from India. Both satellites will feature an L-Band uplink capability as well as a camera system built by Virginia Tech. RadFXSat-2/Fox-1E is part of the ELaNa-XX launch and features a new linear transponder now expected to fly in Second Quarter 2018.

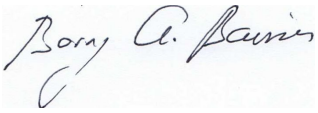
As is AMSAT's tradition, this edition of the Proceedings contains details about our satellite designs that we're placing in the public domain through formal publication. We're releasing these materials so that we may openly discuss our spacecraft development efforts in public as well as our desire to encourage greater use of amateur radio in space, and to provide a foundation for other institutions to utilize our design. Indeed, University of Washington is incorporating the Fox-1E comms package into their DubSat-1. There are other flight opportunities for AMSAT's comm package developing. "An OSCAR in every university CubeSat" will further enhance amateur radio's presence in space while supporting scientific payloads.

Meanwhile, the 35th AMSAT Space Symposium provides an opportunity to discuss our next generation of projects. The ground terminal development work continues as part of ASCENT (Advanced Satellite Communications and Exploration of New Technology). Our goal is to develop an "affordable" ground station for use with a variety of potential amateur spacecraft or payloads using the 5 GHz and 10 GHz amateur bands ("Five and

Dime”). AMSAT collaborated with Ragnorak Industries on their Heimdallr 6U CubeSat design, developing the communications system for this spacecraft that was in competition for a launch opportunity on Exploration Mission-1 (first flight of SLS-1 with an Orion spacecraft) that would place Heimdallr in lunar orbit. Unfortunately, when the announcement was made in August, Heimdallr was not one of the three primary spacecraft NASA selected for EM-1. Such a communications package design, however could potentially be used on an AMSAT-built HEO satellite should such an opportunity arise. The Ground Terminal design would also be compatible with an amateur payload developed by Virginia Tech should there be an opportunity to fly such a payload attached to a future geostationary or geosynchronous satellite. As the Fox-1 program comes to a conclusion, AMSAT has started conversations about our next project, designated “Golf” that also represents “Greater Orbit Larger Footprint”. Stay tuned to see how this new opportunity develops.

After forty-eight years, we continue to develop remarkable satellites with educational outreach and scientific benefit that provides the basis for fulfilling AMSAT’s mission. Your support will help to ensure a bright future.

73,



Barry A. Baines, WD4ASW
President

Frank Robert “Robin” Haighton, VE3FRH, of Burlington, Ontario, passed away on December 2, 2016. He was 79.

Robin served as AMSAT-NA president from 2000 until 2004. Prior to that, he was an AMSAT executive vice president and a long-time member of the Board of Directors. During his tenure as AMSAT president and Board member, Robin guided the organization through the launch and subsequent anomalies with the Phase 3D (AO-40) satellite — the most expensive and elaborate amateur satellite project in history. He was also instrumental in the birth, development and launch of AMSAT-NA's AO-51 satellite.

In 1997, Robin hosted the AMSAT Space Symposium in Toronto, Ontario, the first time that meeting was held outside of the USA. At about that same time, Robin stepped up to the plate to become AMSAT's Canadian Liaison, working a number of AMSAT issues north of the US/Canadian border. Robin coordinated AMSAT's consultant role with the Canadian MOST (Microvariability and Oscillations of STars) satellite project. Through his tireless, "hands on" efforts with the satellite's Canadian builders, AMSAT was able to bring critical analysis and mentoring skills to bear on the project which also garnered a large monetary donation to AMSAT as a result. MOST was successfully launched in 2003 and the satellite remains on orbit and largely operational to this day.

As a founding member of the Amateur Radio on the International Space Station (ARISS) project, he was one of two international delegates from Canada. Robin contributed significantly to ARISS through his ideas, guidance, and wise counsel. He challenged members of the team to look at things in a different perspective. As a result of his sage advice, they were able to work through these issues and arrive at a common approach. Through his responsibilities as AMSAT President, he made sure that we were able to develop and deliver the ARISS hardware systems and support the technical mentoring of the schools and local hams. Robin quietly provided outstanding leadership that enabled ARISS to grow and evolve into the outstanding ham radio educational outreach program that it is today.

Robin was a Life Member of Radio Amateurs of Canada, a member of ARRL and a member and past president of the Burlington Amateur Radio Club. Born in England, he emigrated to Canada with his wife Diana and son Andrew in 1965, his son Peter was born in Canada. A radio amateur since 1977, he held the call signs VE3FRH, VE3IUI, and GD4INU.

His legacy will live on both within the hearts and minds of children and family members that experience ARISS, as well as those who served with him during his many years as a senior officer of AMSAT.

May your spirit soar among the stars!

Keith Baker, KB1SF/VA3KSF, Frank Bauer, KA3HDO, Lee Hatfield, K2HAT, Alan Griffin

Pat Gowen, G3IOR, of Norwich, Norfolk, UK, passed away on August 17 after a long illness. He was 85. The amateur-satellite world has lost one of its true pioneers. Pat was a co-founder of AMSAT-UK, and was elected to the board of directors of AMSAT-NA in 1974. He was also the longtime satellite columnist of Practical Wireless, as well as a frequent contributor to The AMSAT Journal and Oscar News. He was the first to work 100 DXCC entities via satellite (all LEOs), and received Satellite DXCC No. 4 when his QSL cards arrived. Fluent in Russian, Pat enjoyed conversing with the cosmonauts aboard MIR and ISS.

Martin Sweeting, G3YJO, Chairman of AMSAT-UK and founder of Surrey Satellite Technology, Ltd. (SSTL) remembered Pat this way, "If my memory serves me well, I first met Pat G3IOR sometime in the mid-1970's when I became interested in space and naturally amateur radio satellites - starting with OSCAR-6. In 1975, Pat was a founding member and sometime Chairman of AMSAT-UK along with other stalwarts such as Roy Stevens G2BVN, Arthur Gee G2UK and Richard Limebear G3RWL and had a strong interest in not only the OSCAR series but also especially the RS satellites from the then Soviet Union. Starting in late 1973, Pat produced the first editions of OSCAR News as a typed newsletter providing essential orbit tracking information for the new OSCAR-6 satellite (at a time before easy access via the internet!) and contributed more timely updates and observations via the AMSAT-UK 80-metre net each Sunday morning - along with a news round-up each month transmitted by G3RWL. Pat attended and contributed to the early AMSAT-UK Colloquia at Surrey.

Pat surprised the AMSAT community in 2002 by detecting CW signals from the long defunct OSCAR-7 satellite whose batteries had failed in 1981 but came back to life after the batteries eventually went open circuit enabling operations in sunlight which conntinue to this day, making OSCAR-7 the oldest functioning satellite in Earth orbit. On June 21, 2002, Pat posted a message on the amsat-bb, "I have just come across something most remarkable this Friday 21st June evening. Checking out interlopers in our 145.800 - 146.000 MHz space band with a new vertical now atop my 60' tower and working like magic, at 1728UTC I came across a beacon at S.7 sending slow 8 -10 wpm CW on 145.973.8 MHz. It

slowly Dopplered down to 145.970 MHz before going out at 1739 UTC."

In addition to his work with AMSAT and satellites, Pat was a prominent HF operator and DXer. He had confirmed QSOs with all DXCC entities and had also been a member of the First Class CW Operators Club (FOC), as well as a leading member of the Norfolk Amateur Radio Club. Literally hundreds of new amateurs obtained their licenses as a result of Pat's NARC training classes. Pat was passionate about amateur radio, amateur satellites and the environment; he contributed much over his long life to each of these and will be sadly missed.

Professionally, Pat was a biochemist, retired from the University of East Anglia. He had also served as a magistrate in the Norfolk courts, the youngest in England at the time. No mention of Pat would be complete without noting his decades of work in cleaning up the beaches and waters of his beloved Norfolk. He leaves his wife Norma, a noted landscape artist.

Ray Soifer, W2RS

# Ascent Ground and Satellite Demonstration

**By Ray Roberge (WA1CYB) & Howie DeFelice (AB2S)**

Amateur radio operators have not had a high earth orbit (HEO) satellite since the untimely loss of AO-40 in 2004. Allot has changed in the satellite industry since then. It took a multi-national team six years to design and build AO-40 at a cost of several million dollars. It weighed in at 632 Kg, was about 2 meters across and could generate up to 600 Watts of solar power. In the current environment, the most likely affordable launches will be 3U CubeSat's that are about 30 cm across at the longest dimension, have a mass of about 9 Kg and might be able to generate 50Watts of solar power on a good day. Add to this the government mandated restrictions that all but eliminate international cooperation and the picture looks pretty bleak. AMSAT needed a way to take all these lemons and mix up a big batch of lemonade. This is one of the reasons AMSAT formed the ASCENT team, Advanced Satellite Communications and Exploration of New Technology.

The ultimate goal is to place more capable satellites into higher orbits that give more users more access for longer periods of time. This objective requires multiple disciplines working on many fronts to uncover and capitalize on not only new technology but new launch opportunities and new strategic partnerships with industry and academia as well.

One of the first technologies the team turned to was software defined radio and digital modulation. As with most new things, this was more evolutionary than revolutionary. ArrisSat already used software defined radio techniques in its transponder and much of this technology was used and improved in the Fox series of satellites. To get to a true multi-user transponder that can accommodate hundreds of users within a very small footprint with limited power requires a big step up in complexity in terms of digital signal processing capabilities and spacecraft support systems like power generation, attitude control and propulsion. Fortunately, we don't need to make this giant leap in one step. This paper/demo proposes the first step with a new type of programmable transponder that supports multiple linear and non-linear (FM) conversations simultaneously in a single transponder. Each input frequency 'slice' will have a constant power weighting feeding the downlink transmitter. The downlink will be in X band (10.45 GHz.). The demonstration ground terminal will be low cost and incorporate real time Doppler compensation.

**Be Flexible:**

2 months ago our main goal was to support Ragnarok Industries for its Heimdallr Mission (6U traveling past the moon and returning to orbit the moon). We were not selected for the final 3 slots available. A suggestion was made to put a package on a 3U with a thruster to increase its orbit. We would provide the "science" ala radio based. This demonstration hardware is a concept of what we could do. This paper is written in bullet form since it is a concept demonstration and more words seemed like a waste of time.

**Current Focus Goals:**

- C-Band Uplink, X-Band Downlink (five and dime)
- Provide Frequency/Time Locking Capability for the Ground Station
- Utilize SDR technology to maximize flexibility to Experiment and Modify Operation
- Optimize Satellite architecture to minimize Ground Station costs

- Provide Output maximum Power Weighting for the downlink to prevent satellite capture
- UHF/VHF/L-Band/S-Band/HF optional Uplinks
- Provide a Digital Downlink of Satellite Data
- Provide a Digital Downlink of ID, Time stamp, Location (ala GPS/Grid Square) and uploaded TLE
- Provide for a DVB-S2 Input & Output when available
- Build in Redundant paths where feasible
- Provide multiple Digital Downlink paths
- Provide multiple Voice Downlink paths with different modulation types
- Provide a Multi-channel In- Single Channel out for contact Initiation and/or emergency channel
- Provide a controlled Ham Band Scanner Survey Downlink Capability

**Approach:**

- Build and Demonstrate an example Ground and Satellite system that gives the user a sense of what a modern satellite system might look and feel like
- Concentrate on the system, not the implementation
- Incorporate the latest technology that is available and cost effective
- Use hardware and software that is identical or mimics planned future systems
- Make the communication architecture programmable to facilitate experimentation and optimization
- Use COTS SDR(s) likely to be used to facilitate early launch opportunities

**The System:**

- One Laptop operating as the Satellite Processor and Display. Connected to 2 SDR receiver channels and 1 downlink transmit channel. See Figure 1
- One Laptop operating as the Ground Station Computer and Display. Connected to 2 SDR receiver channels and 1 uplink transmit channel. One of the receiver channels is strictly for demonstration of a local input, in place of a microphone.

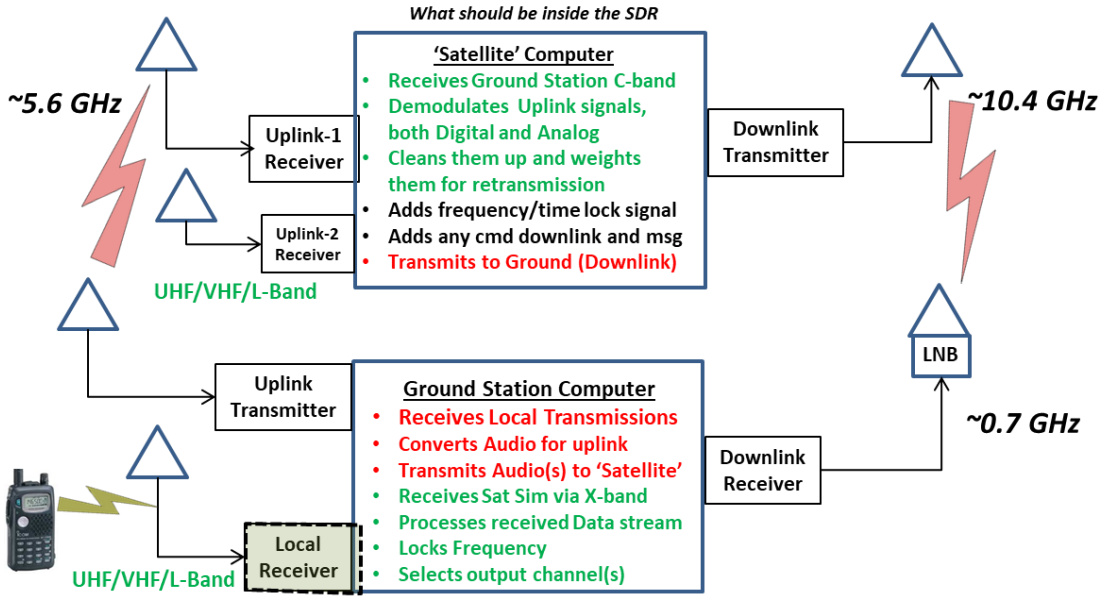



**Figure 1 -Basic Communications System Functions**

**Satellite Communication Processing:**

- Requirement  is that all ground stations have locked onto the satellites beacon and all transmit uplinks will have Zero Doppler at the satellites input
- Satellite Transmits both a CW beacon and a phase modulated 1 pulse per second timing pulse
- All receivers (2 in this case) are translated to wideband baseband for channel separation
- Input frequencies are shifted and filtered according to the programmable configuration plan
- Each segment of the spectrum (channel) is handled in accordance with its intended modulation use
- All Channels have a separate filtering, AGC and squelch circuity set in accordance with its intended modulation use. (AGC timing and Squelch thresholds are different with mode)
- Channels can be either linear, regenerated or cross-modulated (ex: NBFM in, USB out)
- Channel content can be analog or digital
- **Control Ground station can upload a new channel allocation and processing**
- Figure 2 shows simplified example channel processing

**GNU Radio Details**:

- Outer Layer of the flow graph has a Pseudo Doppler insertion (input and output) to simulate Satellite Doppler at 3 selected altitudes generating +/-250KHz,+/-150kHz and +/-50kHz with a 10 min pass time
- Receiver sampling rate minimum is set by the maximum Doppler expected plus the bandwidth of the communication spectrum. In this case +/- 250 KHz Doppler and 256 KHz Output Spectrum, so ~2**20 (1,048,576Hz) was selected +/- depending on the SDR used
- Input receiver stream has a frequency offset to shift the IQ imbalance/DC term by ~65kHz
- nbfm Channel is split into 2 nbfm channels with less deviation for no good reason, just experimented with it, besides who doesn't want more channels!
- The base Squelch level is set by the receiver tab. Individual channel levels add or subtract from this level
- mseq_substitute_15.py is a 2**15 bit pseudo random time/ranging waveform heir block. The ground receiver cross correlates this into one PPS (like GPS). It's weighted lower in the output power spectrum since it has 40 dB of processing gain
- The channel filters are definitely not optimized, but are a 1st approximation of the desired response
- The Wideband Digital channel is a linear channel for this Demo as a place holder.
- Hieratical blocks were not used in most of the flow diagram so as to observe the system complexity, See Figure 3
- No python or C++ programming was used to show what can be done with standard GNU Radio
- The demo system has several Tabs and QT displays that would be disabled in a real system since it eats up lots of resources and no one is there to see the display!

You can download the concept demonstration system from here:
https://github.com/WA1CYB/satellite_ground_emulator/tree/master/Ascent/Concept%20Demo%20System

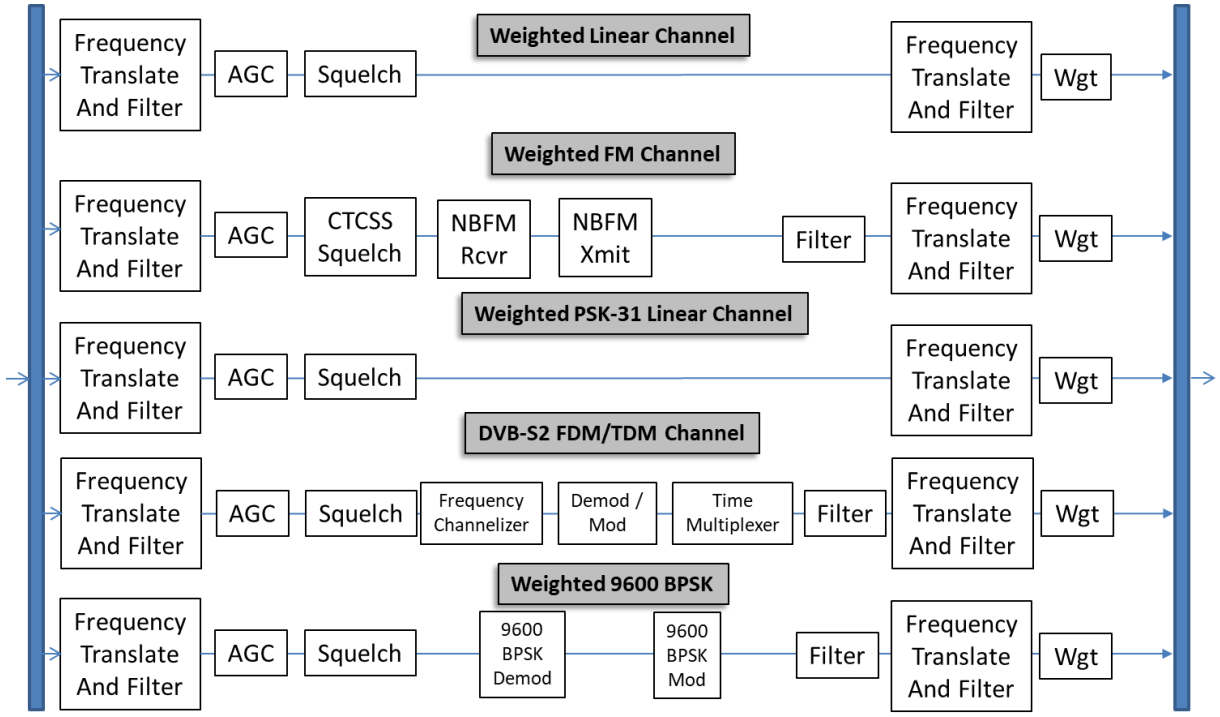



**Figure 2 –Satellite Channel Processing Examples**

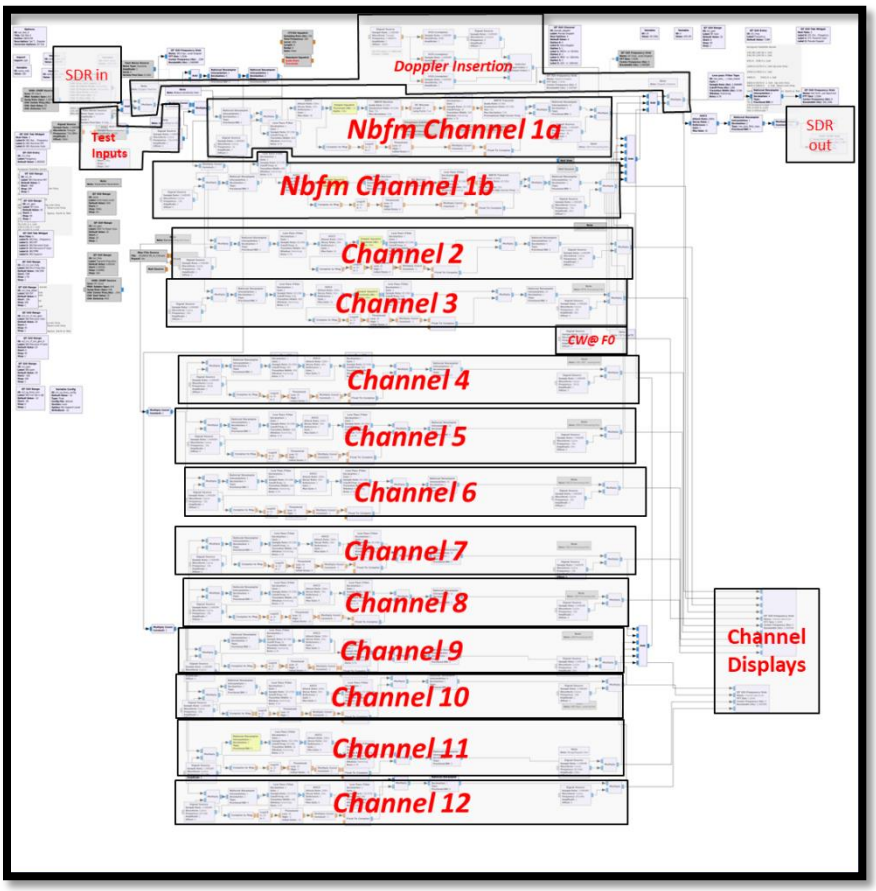



**Figure 3 - Typical GNU Radio Flow Graph – Not Pretty**

**Why Programmable? :**

- Because we can! That's the beauty of Software Defined Radios. We can launch with one band plan and set of mode possibilities and later reconfigure the satellite for new modes and band plans.
- We can refocus the satellite resources to assist in emergency situations. For instance digital messaging may be more important than voice. Or a chat with psk31 should be the highest priority and weighting to get the real time status message through with minimal ground equipment, etc.
- It enables a common satellite method that can be used and programmed differently before and after launch
- It enables experiment possibilities of cross band and cross modulation type communication
- Can have selected receive frequencies that are polled on a specific time schedule for emergency use

**"Science" Experiment Possibilities**:

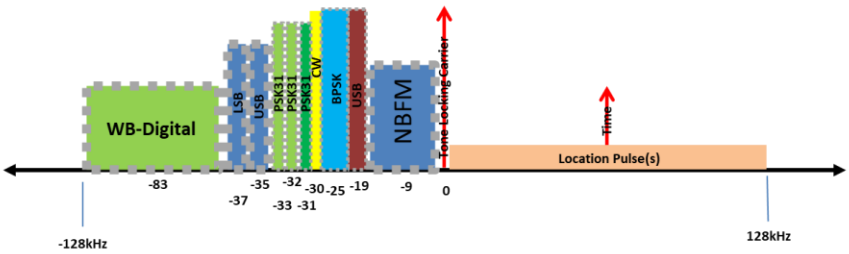
The second receiver can be tasked to do other functions while the main one continues on. The possibilities could be to conduct an Amateur Frequency Survey (multi-bands). Examine propagation via WSPR and WJT modes, upper atmosphere scattering and urban noise sources versus frequency etc. Or, multiple frequency inputs on different bands with common output band channel (Cross band input) for emergency use. Or, Shared Aperture Antenna Steering or Automatic logging of satellite users/peak signal level /vs satellite location or Very weak satellite transmitter power beacon experiments (no power amplifier, straight SDR power) or even more satellite input frequency receivers (1 chip for each 2 frequency bands) or Harmonic beacons for frequencies above x-band…… endless possibilities!

**What can we build now you say? :**

Everything! (Although it may not work completely just yet). We have provided the framework to simulate the satellite and ground nodes using laptops for the processor, as opposed to the built in SDR processor in some SDRs. Since the ETTUS SDR runs GNU Radio, as does the laptop, its porting should be an easier step than coding its FPGA directly. Using the FPGA makes the system faster and more reliable, so in the end, that's the direction we should go. (My opinion of course).

**The demo system:**

Figure 4 shows the selected Band plan to demonstrate the concept. It has a dozen functions covering a potential bandwidth of +/- 128 kHz. The satellites output power is set by the combination of the dozen functions. Some functions like NBFM have a constant peak power while others like Lower Sideband have an average power much lower than its peak so to equalize the energy used one can enhance the LSB at the expense of the NBFM. Also the narrow band PSK31 channels are assumed to be higher priority so they are given more of the output energy. Since the entire band plan is predicated on receiving the Tone Locking Carrier, it was given the most power allocation. The location and 1 PPS timing was given less peak power because of the high processing gain.




1. Tone Locking Carrier: Enables Ground Lock of Spectrum when Doppler is present
2. NBFM: Voice Channel(s) Translated from a Low Band
3. USB: Combined Multi-channel In Initiate contact and/or emergency channel down
4. BPSK: ID, Telemetry and data stream from satellite
5. CW: CW Bandwidth(s) Translated from a Low Band
6. PSK31-3: Satellite broadcast of ID, Time stamp, Location (ala GPS) and TLE
7. PSK31-2: Chat Channel #2
8. PSK31-1: Chat Channel #1
9. USB: Channel #3
10. LSB : Channel #4
11. Wide Band Digital: Channel #5
12. Location Pulse(s): Enables Precise Ranging/Location

**Figure 4 – Demonstration Downlink Frequency Band Plan- Arbitrary Peak Weighting**

Figure 6 shows the Satellite control program output spectrum when one satellite SDR receiver was tuned to 2 meters and the second one was tuned to an ATSC TV station calibration frequency. This screenshot was not from the current flow graph, but one developed on the way. Every 2nd frequency block allocation was attenuated so as to see the band plan. The lower right area shows how one could reallocate the blocks, weighting and frequency offsets and filter bandwidths.

**Figure 5 -2 Satellite Receivers Enabled, 1 tuned to 2 meters, the second tuned to a ATSC HD pilot tone**

Some may rightly question why even put USB/LSB on the potential modulation types for the channels. Keep in mind even WSJT uses USB to gather it's data for further processing. The ground receivers SDR output can be re-directed to a virtual audio cable which in turn can be picked up by FLDIGI, WSJT-X, other programs we have or remote operation for further processing if desired.

Figure 6 shows the Satellite set up to receive in C-Band for Satellite Receiver #1 and receive in the 432MHz band for the second receiver. The output from the SDR is at C-band, which is then mixed up to X-band for transmission to the Ground Station. Figure 7 shows a test received spectrum using SDR-Console by Simon Brown.

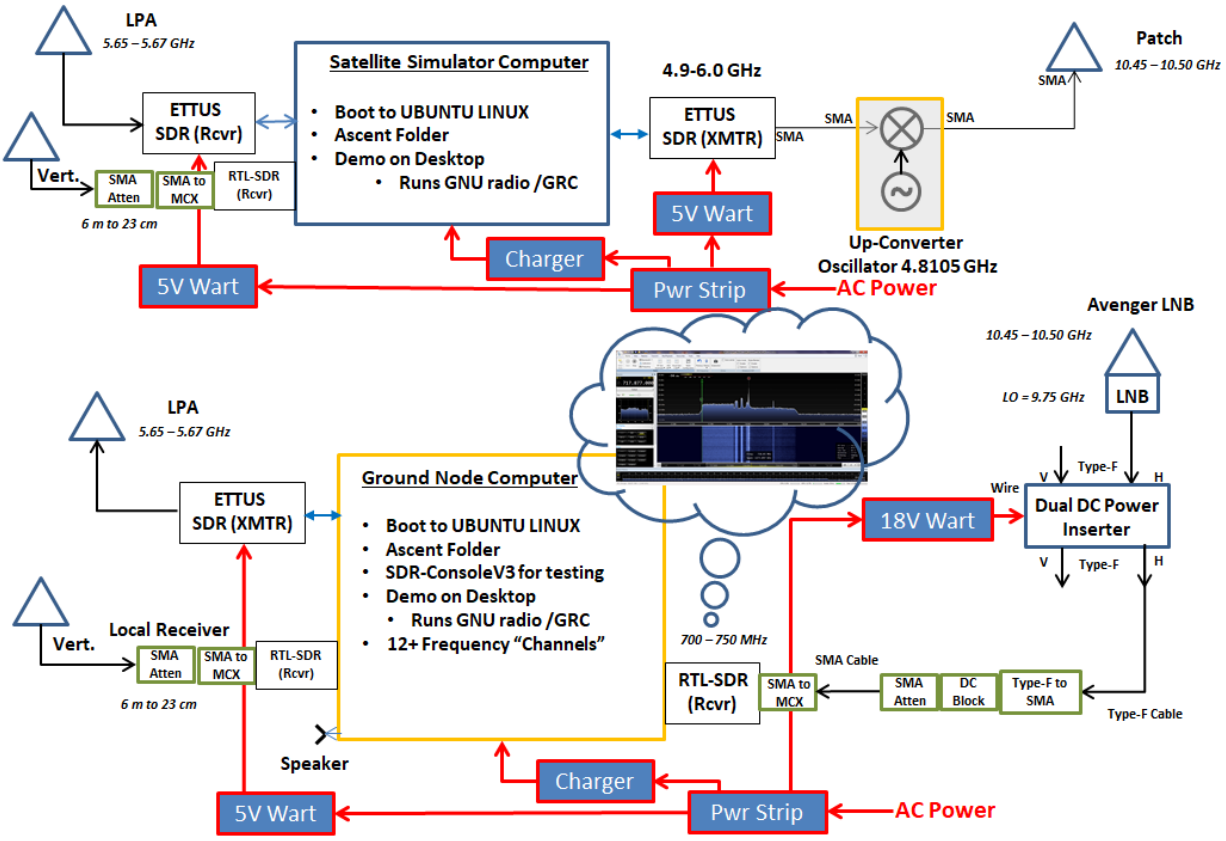



**Figure 6 -C-Band and 70 cm band received uplinks with common X-Band downlink**

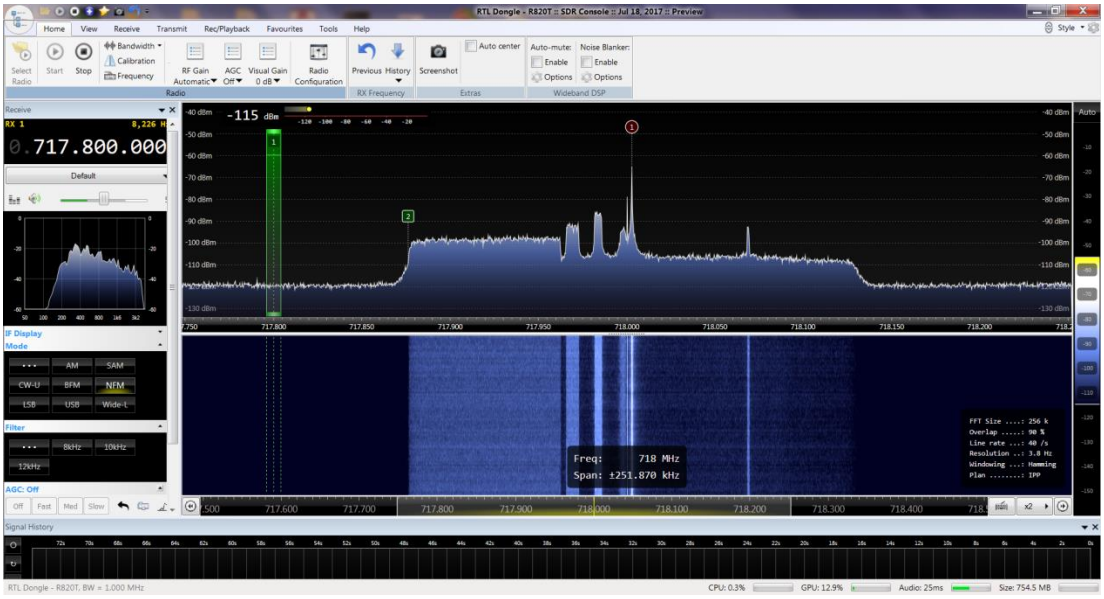



**Figure 7 – Received (noise channel inputs) test spectrum- every other channel suppressed**

**The Ground station:**

Figure 8 shows the functional diagram of the barebones demonstration Ground station. A second SDR (RTL-Dongle) was added so as to have a local input (instead of a microphone) that people could utilize locally at the symposium for a input (doesn't everybody carry a HT?). The ground station has recorded uplink messages internally generated. The received signal consists of a chat channel window and a single audio window. Not the prettiest system, but hopefully it gets the point across.

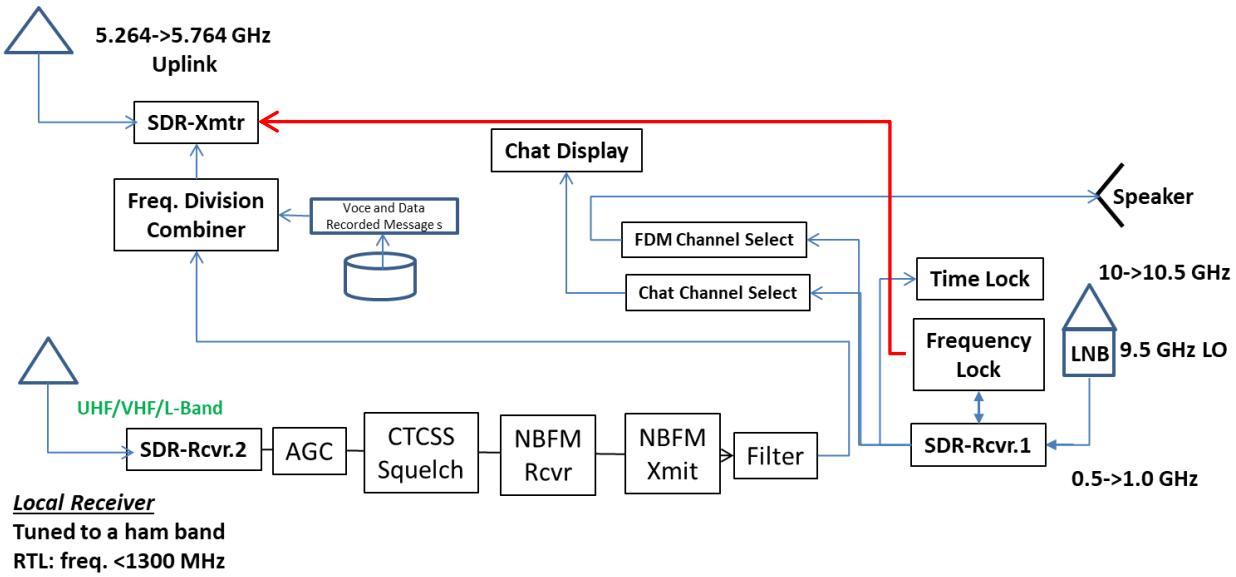



**Figure 8 – Simplified Bare Bones Ground Station**

**Conclusion:** We can build it if the opporunity arrises. Flexable programable digital signal processing allows for multiple uses and science experiments.

# Frequency Calibration for SDRs - Without GPS

**By Ray Roberge, WA1CYB**

**Summary**: I will describe a program that commands your Software Defined Radio to frequency precise TV signals. By looking at your local TV signals, calibration to less than 100 Hz at 600 MHz is easily attainable even with an inexpensive RTL-SDR Dongle.

**Perceived AMSAT Need**: To make a satellite/terrestrial ground station as inexpensive as possible it is probably necessary to use some parts that have significant frequency error for use with the 5 GHz and 10 GHz bands. This would include the transmitter frequency determining components, baseband frequency source and all the mixers used to up-convert the baseband to 5->6 GHz (C-Band). Similarly, on the receive side we need to down-convert the signal to baseband with the possibility of intermediate conversions in between. In the end to handle the modulation it is envisioned that a software defined radio will be needed to demodulate the received satellite data/audio/beacon etc.

**Solutions:** The easiest way and most expensive way to handle the frequency uncertainty is to have an atomic frequency source ($$$). The second best method is to have a GPS Disciplined Oscillator. With a moderate cost, this might be a solution for many hams. The third method is to have a crystal oscillator and a way to calibrate its frequency. This being the least expensive path provides the reasoning for an inexpensive calibration source. I put together a program (free) so that this uncertainty can be minimized. This tool can be easily modified under GNU Radio Companion for other SDRs or used as is.

**Different Methods:** At HF you could use WWV, WWVH, CHU or the Russian frequency and time broadcasts. If your source is low frequency you could compare it to them and get some measure of accuracy. Since we need a way to calibrate a signal at significantly larger frequencies than the ones at HF, this can be difficult. You could use the harmonics of your oscillator by making a comb generator. If your crystal oscillator is portable and fairly stable with frequency, a local club can provide someone with access to a stable source to compare it to. An additional method is to use a source that is broadcast on UHF. In the US it is the ATSC, high definition TV, signals. These signals use multiple frequency tones to broadcast the picture and audio via digital encoding. For their scheme to work, they need a reference frequency for the TV's AFC to lock onto. To prevent adjacent channel interference the frequency has to be common with its neighbor. This pilot tone frequency is what we exploit. The mandated FCC frequency specification for the TV stations is no longer required, but most stations have designs that predate that. Many have Atomic Clocks (Cesium or Rubidium). They also use a GPS Disciplined Oscillator if not an Atomic clock. Now what remains is to find the pilot tones and see where your receiver says they are. You can compare the two to determine the frequency error. By looking at several stations you can determine your frequency error versus frequency. Once you know that, you can correct the slope of that error with your Software Defined Radio ppm setting. There will then be some residual error because of the capability to set the ppm accurately is limited. We can measure the delta.

**Why is UHF calibration good enough?:** Using a Block Low Noise Down-Converter like an AVENGER, with its output at UHF makes it a good fit for this calibration technique. If you have a direct down conversion receiver, from 10GHz to baseband this program won't help very much. If your SDR has a wide band operating capability using the latest 70MHz to 6GHz chip sets, then you can also utilize this to calibrate your receiver also.

**Pilot tone –Technical:** For those interested in the basis for the pilot tone, here are a few quotes and references:

**1)** **"**Consequently, the pilot frequencies of all transmitters in a network shall be maintained within ±1/2 Hz of nominal frequency (i.e., ±1 Hz of one another)." *"ATSC Standard for Transmitter Synchronization, Doc. A/110:2011, 8 April 2011 page 74.

**2)** The formula for determining the location of the pilot with respect to the lower edge of the theoretical occupied bandwidth is:

$$\text{pilot frequency} = \left[ 6 - \frac{Tr \times \frac{208}{188} \times \frac{313}{312}}{4} \right] \div 2 \text{ MHz} \quad = 309440.55944056 \text{ Hz (I used 309441.Hz)}$$

Tr is the Transport bit rate, Tr (approximately 19.39 Mbps) in a 6 MHz channel

From; ATSC Digital Television Standard – Part 2: RF/Transmission System Characteristics, Doc. A/53 Part 2: 2011, 15 December 2011

**3)** A more precise Tr is:

$$T_r = 2 \times \left( \frac{188}{208} \right) \left( \frac{312}{313} \right) \left( \frac{684}{286} \right) \times 4.5 = 19.3926584597511 \text{ Mbps}$$

From: ATSC: "Digital Television Standard, Part 3 – Service Multiplex and Transport Subsystem Characteristics," Doc. A/53, Part 3:2009, Advanced Television Systems Committee, Washington, D.C., 7 August 2009.

This means that every 6 MHz, starting at 470,309440.55944056 Hz to 686.309… MHz you ***might*** have a constant CW tone to lock onto and calibrate your system, if you have TV reception in your area.

**The Program:** The goal was to provide a spectrum plot and a waterfall plot of the ATSC pilot carriers. GNU Radio provided an easy way to implement a calibration program. *Figure 1* shows the Block Diagram of the receiver. The SDR receiver that I targeted was a simple, inexpensive RTL-SDR Dongle. Its frequency error was initially high but can be corrected to be less than 100 Hertz at 600 MHz. GNU Radio is best run under the Linux operating system, but can also run under windows. Here is a link to the program.

https://github.com/WA1CYB/satellite_ground_emulator/tree/master/Ascent/Frequency%20Calibration

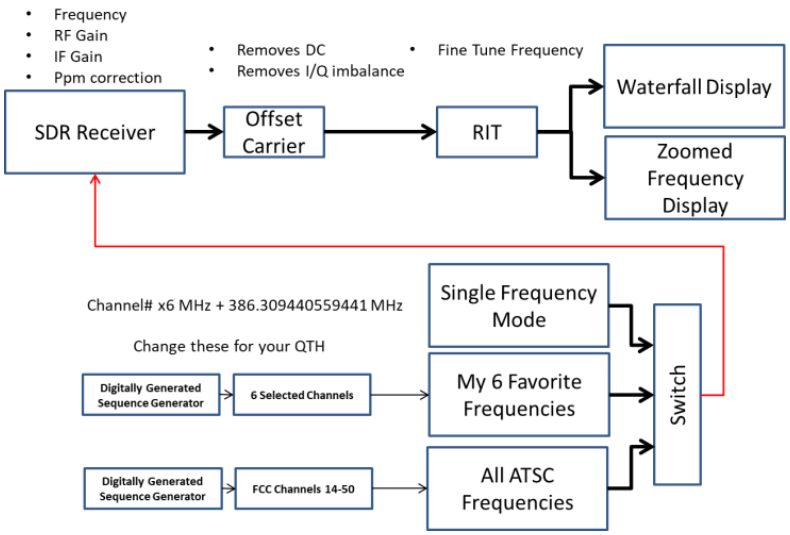



**Figure 1 - Frequency Calibration Receiver Block Diagram**

*Figure 2* shows a screenshot of the program using a RTL-SDR taken from my QTH in the Boston Area. Ideally the waterfall would have one straight red line as the receiver jumps between the 6 selected frequencies, all centered on zero Hz in the display. Use the *Auto Scale* button and the sliders to optimize the display. You can also change the mode to scan all channels, or just look at one channel.

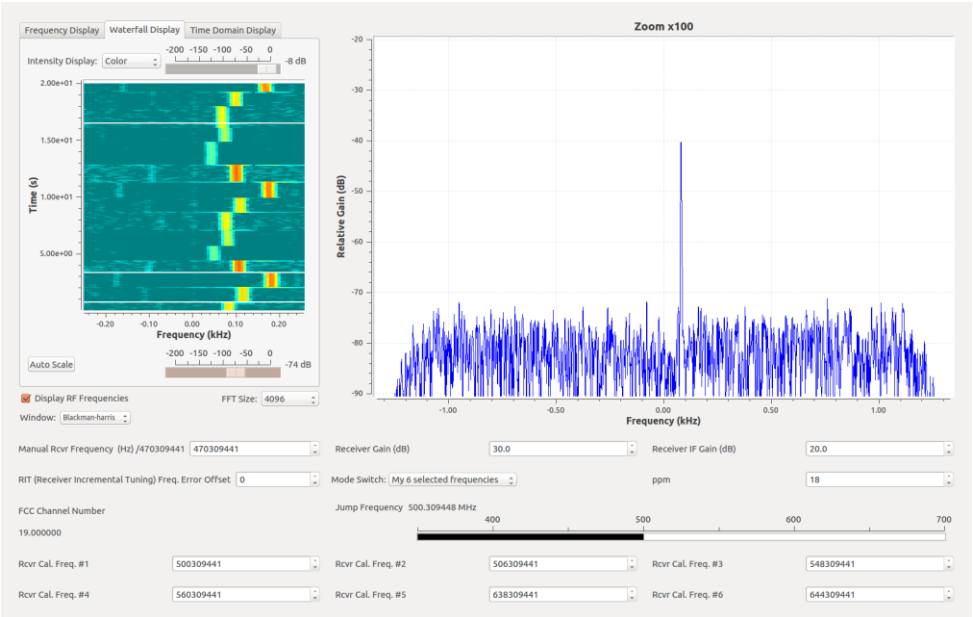



**Figure 2 - RTL-SDR in 6 selected frequency mode, RIT = 0Hz**

The part per million (ppm) setting is the major error source for this RTL-SDR. When the program is first run, use the frequency display tab on the left graph. This covers a wider bandwidth than the displayed frequency on the right. Change the ppm number to get the strongest channel as close as possible to zero Hz. After that, use the right hand display along with the waterfall tab to fine-tune and center the group of selected frequencies as close as possible to zero hertz. To finalize the calibration, use the Receiver Incremental Tuning (RIT) to fine tune the

selected frequencies to the center of the waterfall. With the ppm and RIT values you can optimize your frequency readout in your favorite program.

Changing the selected frequencies is done by using up/down arrows next to the selected frequency box. Note the arrows will increment the value by the channel separation, 6 MHz. Alternatively, you can type in a frequency or copy and paste the frequency in any of your 6 selected values.

*Figure 3* shows a screenshot of the program using an ETTUS N-210, with a GPS-DO with no calibration routines done before the program was run. A much better starting point compared to the RTL-SDR, but at a cost. In my experience other SDRs performance is between these two SDR types.

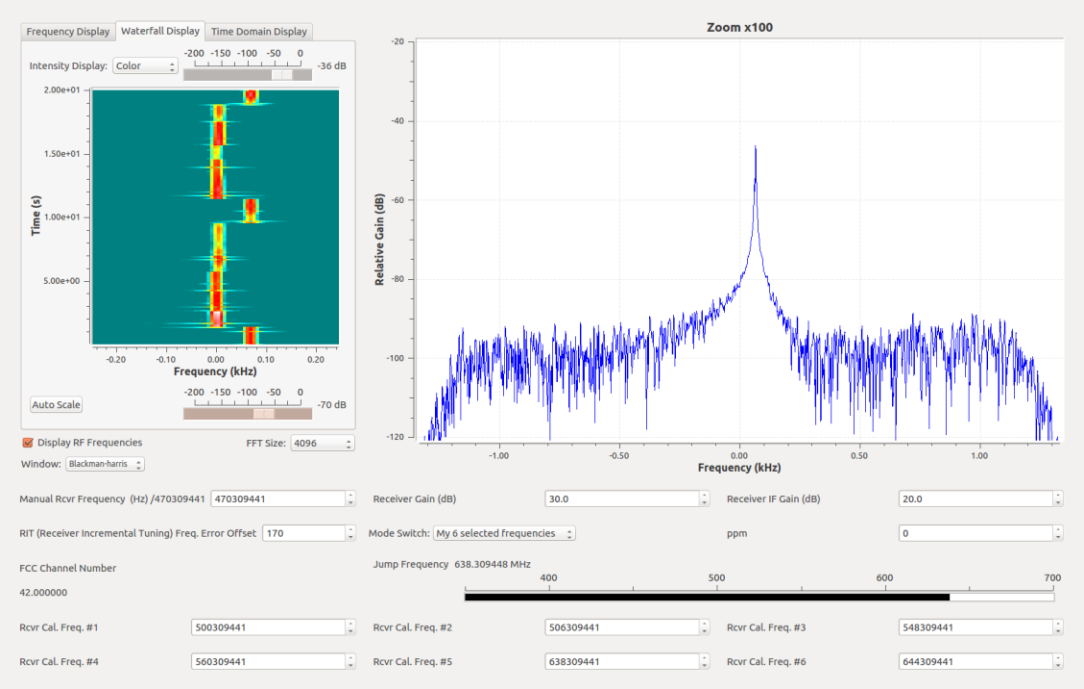



**Figure 3 – ETTUS N-210 (un-calibrated) in 6 selected frequency mode**

For the observant person, you will notice that the displayed jump frequency is not exactly the desired frequency (7 Hz off) even though the correct frequency (within 1 Hz) was sent to the SDR. The reason appears to be the limit of a 32 bit display/command. That's the best I could do with the standard GNU Radio flow graph. Another discovered oddity is the ppm value can only be an integer. The reason is the RTL-SDR driver can take a ppm value with more precision, but the RTL-SDR ignores all but the integer portion. You can also observe the frequency drift of the SDR from a cold start. After 10 to 15 minutes the RTL-SDR is remarkably stable.

**Conclusion:**

The ATSC TV signals in North America can be used to calibrate your SDR receiver with less than 100 Hz error up to the 23 cm band even with an inexpensive RTL-SDR. This tool provides a method to accomplish that. A calibrated SDR receiver makes operating you favorite satellite program easier to get on frequency.

# A FAULT-TOLERANT, REDUNDANT IHU[i] WITH A FREQUENCY AGILE, DIGITAL COMMAND TRANSCEIVER

By Zach Metzinger, N0ZGO
Jonathan Brandenburg, KF5IDY
Bill Reed, NX5R
Jordan Trewitt, KF5COQ
Jesse Marroquin, K5JXM

## SUMMARY

## OBJECTIVE AND OVERVIEW

ASCENT[ii] is the "skunk works" program[iii] of AMSAT intended to investigate and implement technologies for future amateur satellite missions, including both ground-based and space-based technologies. One team under the ASCENT umbrella is designing and developing an IHU for future missions. The initial design for the IHU was driven by several objectives.

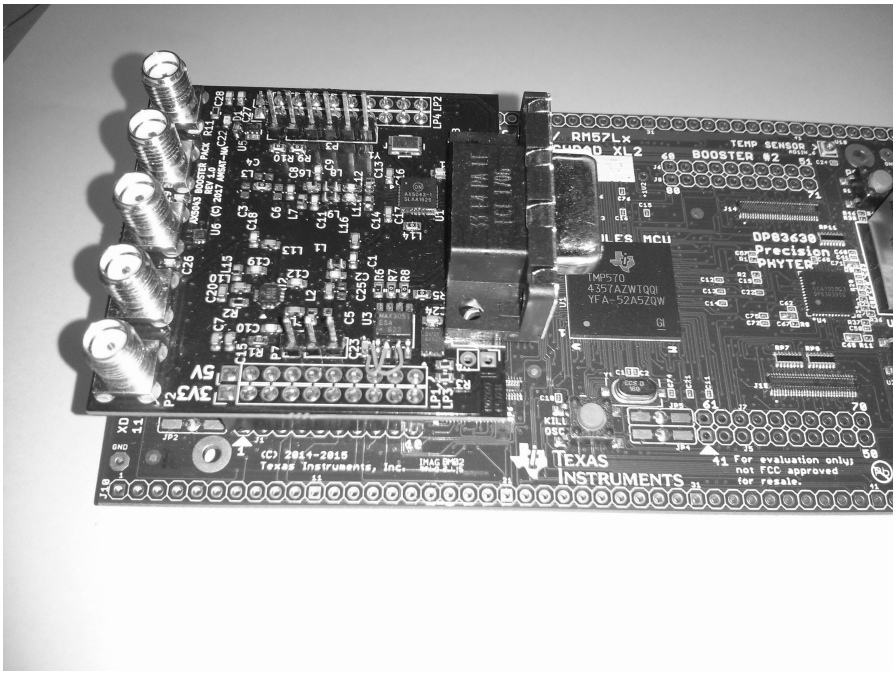



**FIGURE 1 - PICTURE OF AX5043 BOOSTERPACK ON HERCULES DEVELOPMENT BOARD**

### Redundancy

Redundancy in the IHU better tolerates the harsh radiation environment of orbit. Using off-the-shelf components that are not necessarily space-rated significantly reduces the cost of the components but increases the risk of failure. Redundancy in the components is a mechanism to mitigate this risk.

### Autonomous Failover

Autonomous failover allows the IHU to detect certain failures and respond to those failures without requiring explicit ground commands. While autonomous failover increases the complexity of the satellite, it ensures the IHU operates without interruption in the face of temporary or permanent component failures.

### Digital Command Transceiver

A digital command transceiver provides for direct communication and command of the IHU. With this transceiver, the IHU transmits telemetry to the ground or receives commands from the ground without relying on a separate transceiver payload. While the satellite's payload may include a transceiver, the IHU doesn't require a separate transceiver as payload.

### 1U Footprint

By supporting a 1U footprint, which is 10cm x 10cm x 10cm, the IHU is suitable for use on a 1U or larger satellite. There are many 1U and larger satellites being launched and the 1U size can be considered almost a "least common denominator," maximizing the number of launch opportunities.

## Transceiver Board

The team wished to design a transceiver for the IHU that:

- Supports digital transmission standards,
- Has low power requirements,
- Is affordable, and
- Supports a broad range of frequencies.

The AX5043[iv] from ON Semiconductor met these requirements.

## AX5043

Billed as an ultra-low power RF transceiver for 27 to 1050 MHz, the AX5043 modulates and demodulates a variety of digital transmission standards, including GFSK, MSK, FSK, ASK, and PSK[v] with optional forward error correction. This transceiver operates in half-duplex mode, and is typically used for ISM-band data networks.

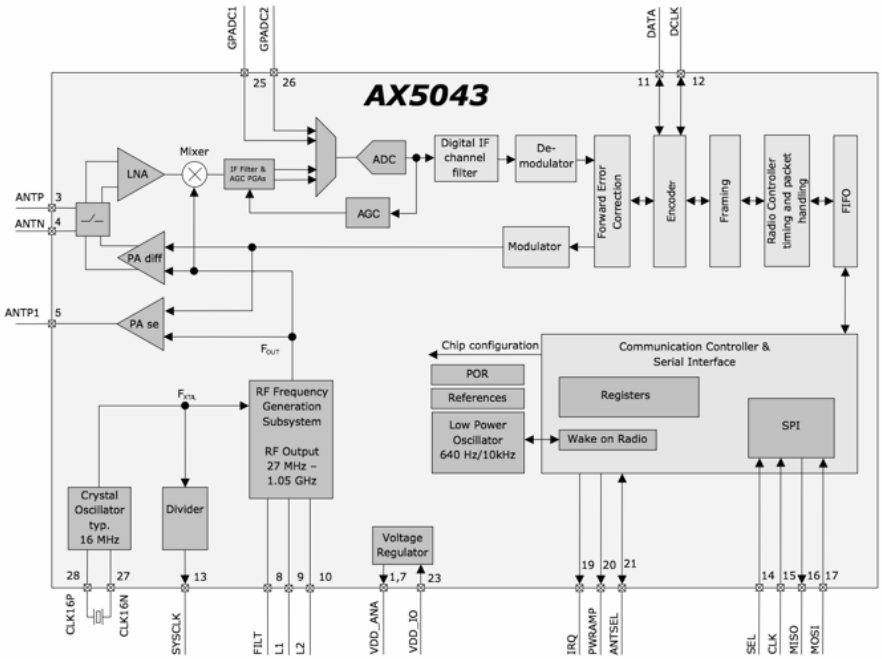



**FIGURE 2 - AX5043 FUNCTIONAL BLOCK DIAGRAM[vi]**

Of particular interest is the AX5043's ability to transmit and receive packets without microcontroller intervention. When transmitting data, the microcontroller provides the data to the AX5043 and the AX5043 takes care of the modulation and transmission. When receiving data, the AX5043 demodulates and decodes the packets then internally buffers the resulting data.

When it comes to power requirements, the AX5043 requires 9.5 mA when receiving and a maximum of 48 mA when transmitting at 16 dBm. There is also a standby mode that requires 500 nA from and the AX5043 supports autonomous wake-on-radio cycling.

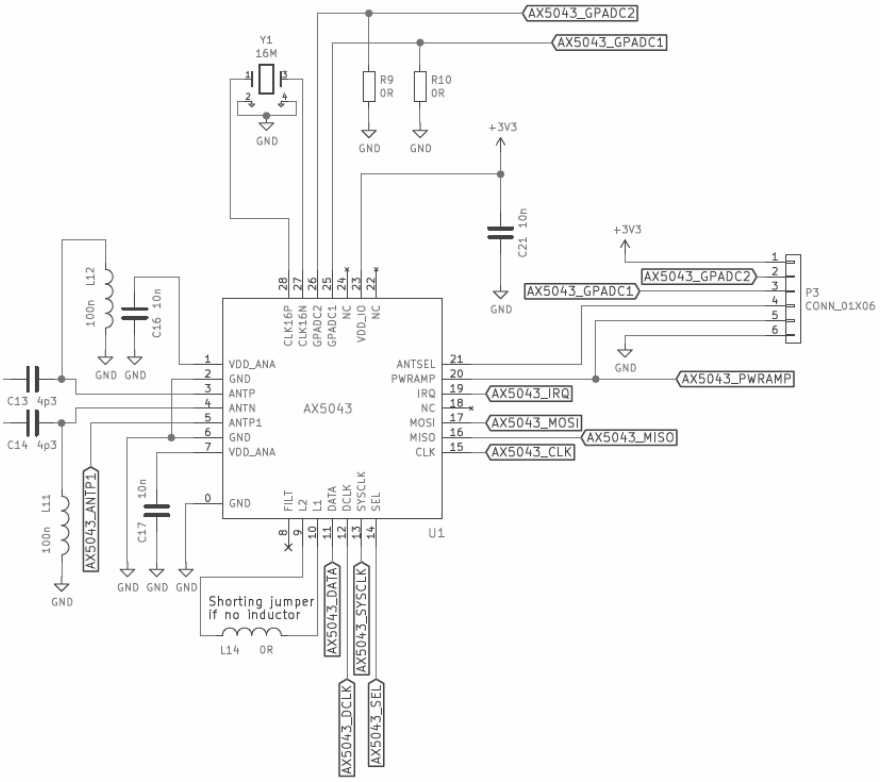



**FIGURE 3 - AX5043 SCHEMATIC ON BOOSTERPACK™**

## FILTER NETWORK

The AX5043 works across a broad frequency range. To assist with designing a filter network to address spurs and harmonics, the data sheet provides typical application diagrams. The initial design of the digital transceiver is based on the use of 433 MHz. 433 MHz is a good compromise frequency as it requires an antenna of modest size and atmospheric attenuation is moderate.

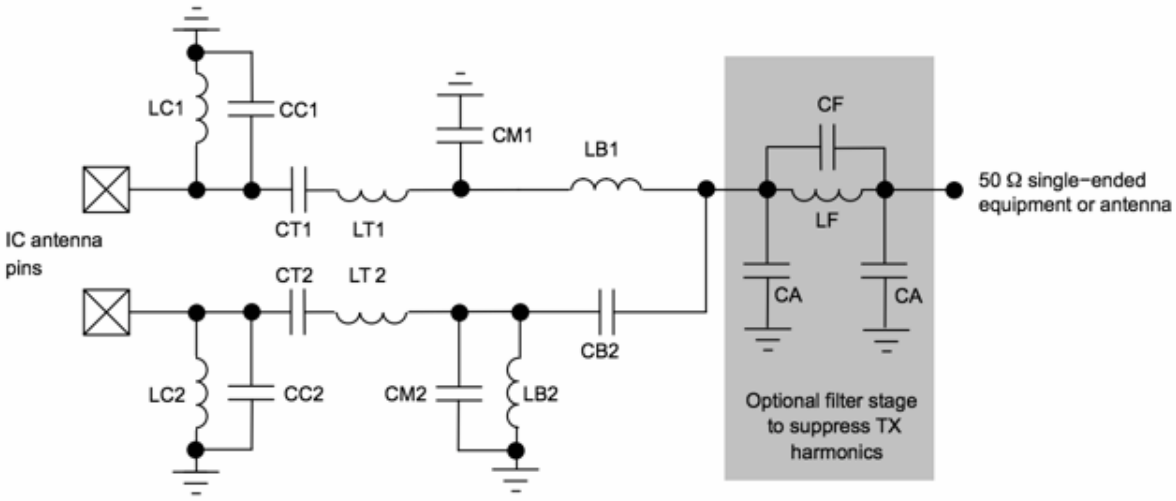



**FIGURE 4 - SAMPLE AX5043 DIFFERENTIAL ANTENNA INTERFACE[vii]**

| Frequency Band | LC1,2 [nH] | CC1,2 [pF] | CT1,2 [pF] | LT1,2 [nH] | CM1 [pF] | CM2 [pF] | LB1,2 [nH] | CB2 [pF] | CF [pF] optional | LF [nH] optional | CA [pF] optional |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 868 / 915 MHz | 18 | nc | 2.7 | 18 | 6.2 | 3.6 | 12 | 2.7 | nc | 0 Ω | nc |
| 433 MHz | 100 | nc | 4.3 | 43 | 11 | 5.6 | 27 | 5.1 | nc | 0 Ω | nc |
| 470 MHz | 100 | nc | 3.9 | 33 | 4.7 | nc | 22 | 4.7 | nc | 0 Ω | nc |
| 169 MHz | 150 | 10 | 10 | 120 | 12 | nc | 68 | 12 | 6.8 | 30 | 27 |

**FIGURE 5 - TYPICAL COMPONENT VALUES FOR DIFFERENTIAL ANTENNA INTERFACE[viii]**

## AMPLIFIER

The AX5043 has a maximum transmission power of 16 dBm at 433 MHz, approximately 40 mW. Any practical use of the AX5043 will require an amplifier. For the AX5043 transceiver board, the MMZ09312B1 2-stage amplifier from NXP Semiconductors performs that function. This amplifier has good specifications including[ix]:

- Frequency range from 400 – 1000 MHz
- Power Gain of approximately 33 dB at 450 MHz
- Maximum power requirement of 500 mA
- Maximum input power of 14 dBm

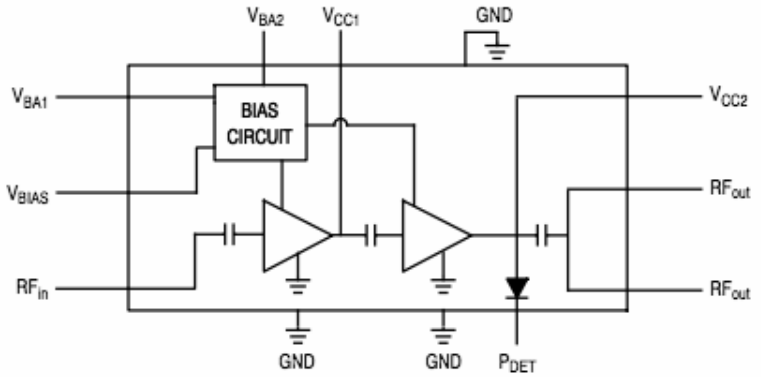



Figure 1. Functional Block Diagram

**FIGURE 6 - MMZ09312BT1 FUNCTIONAL BLOCK DIAGRAM[x]**

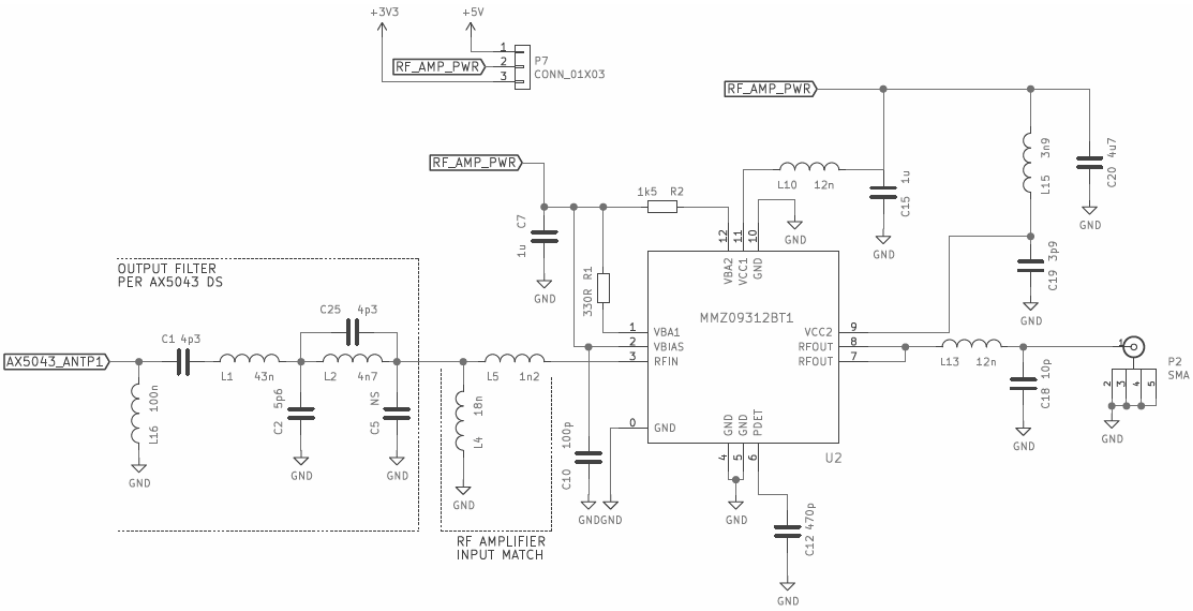



**FIGURE 7 - AMPLIFIER SCHEMATIC ON AX5043 BOOSTERPACK™**

## ANTENNA SWITCH

Should the mission require a separate transmit and receive antenna, the AX5043 transceiver board includes an antenna switch. Currently, a SKY13446-374LF radio frequency switch, from Skyworks Solutions, provides the transmit/receive switching. This switch has typical insertion loss of 0.5 dB and 38 dB isolation from 0.1 – 3.0 GHz[xi].

The team is considering the replacement of this switch with discrete components to reduce the risk of radiation effects. That research is ongoing.

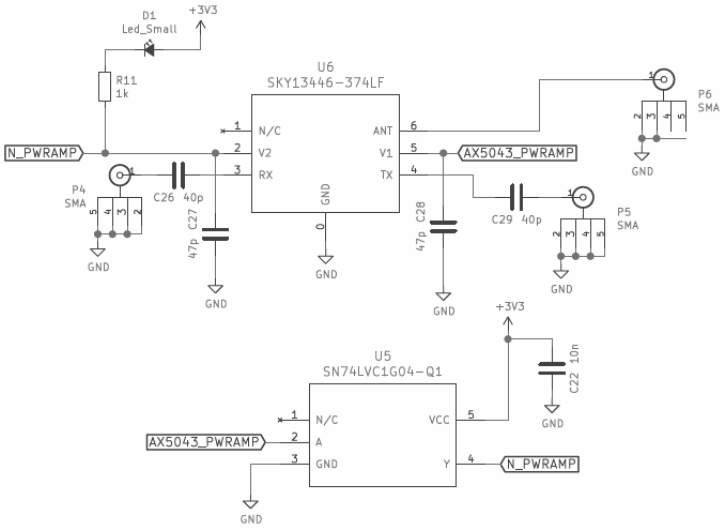



**FIGURE 8 - ANTENNA SWITCH SCHEMATIC ON AX5043 BOOSTERPACK™**

## CAN[xii] INTERFACE

There has recently been increasing discussion regarding the use of a CAN bus on satellites. In part, this interest arises from the common use of a CAN bus in notoriously noisy RF environments such as automobiles. To prototype the use of a CAN bus in the IHU, the AX5043 BoosterPack™ includes the MAX3051 from Maxim Integrated. This CAN interface chip provides the electrical interface between the CAN protocol controller and the bus lines in the controller area network.[xiii]

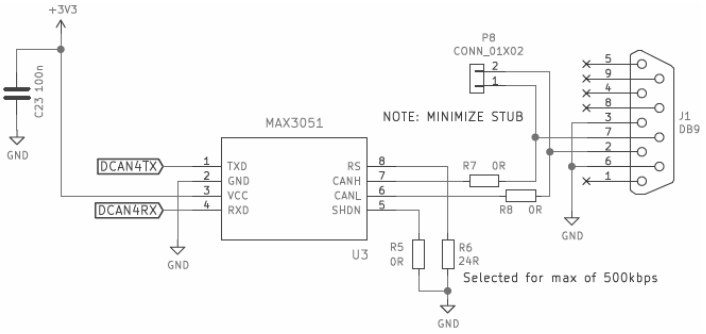



**FIGURE 9 - MAX3015 SCHEMATIC ON AX5043 BOOSTERPACK™**

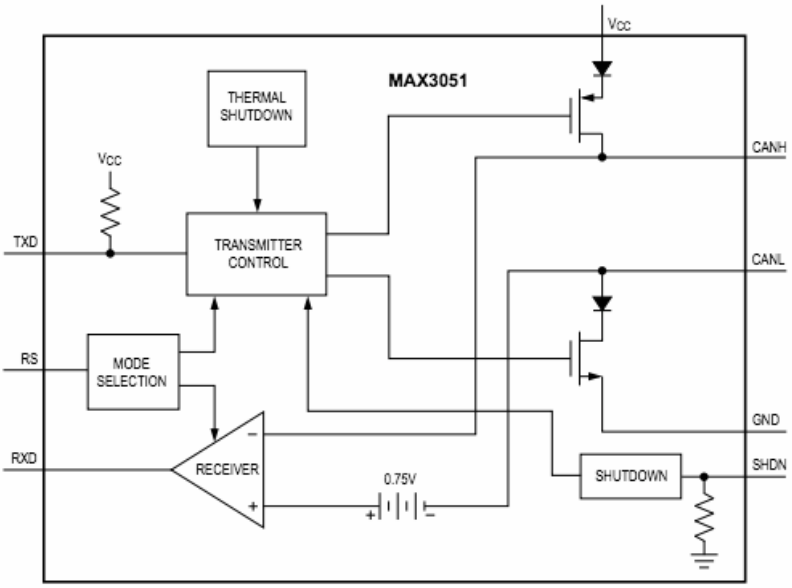



**FIGURE 10 - MAX3051 FUNCTIONAL BLOCK DIAGRAM[xiv]**

## TEMPERATURE SENSOR

The transceiver board also includes an onboard temperature sensor. Not only will this sensor provide useful telemetry when using the AX5043 transceiver board, such as power amplifier temperature, this sensor also provides data for transmission and reception during development and testing. The MAX31725 from Maxim Integrated provides high accuracy +/- 0.5 C readings

with 16-bit accuracy from a range of -40°C to +105°C. Furthermore, the MAX31725 communicates using SPI, a communication interface available on many modern processors.[xv]

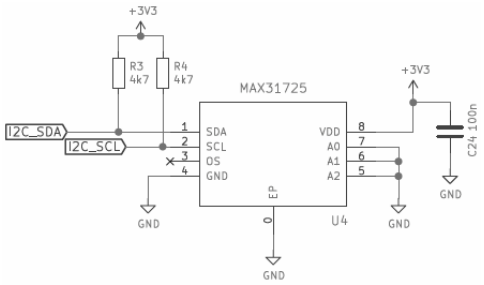



**FIGURE 11 - MAX31725 SCHEMATIC ON AX5043 BOOSTERPACK™**

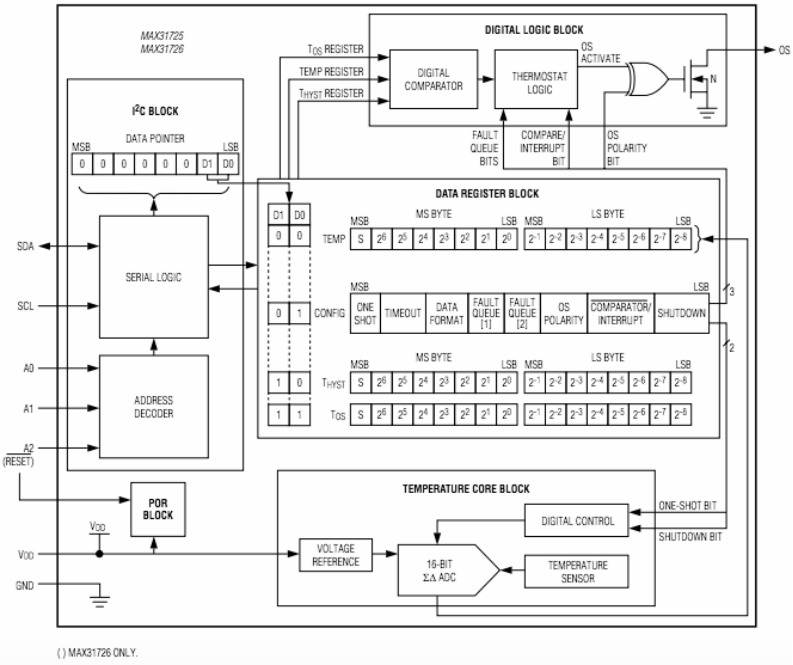



**FIGURE 12 - MAX31725/MAX31726 FUNCTIONAL BLOCK DIAGRAM[xvi]**

## BOOSTERPACK™

The team selected the affordable Texas Instruments Hercules™ TMS570LC43x LaunchPad™ Development Kit[xvii] for prototyping and development. This development kit provides two key benefits:

- The LaunchPad™ Development Kit contains the Texas Instruments™ TMS570 Hercules™ processor. The selection of this processor for the IHU is discussed in the Supervisor Board section later in this paper.
- The BoosterPack™ XL headers allow the team to prototype the AX5043 transceiver board and easily attach the transceiver to the LaunchPad™ Development Kit for testing.

The Texas Instruments BoosterPack™ standard provides a well-defined interface between the LaunchPad™ Development Kit and an add-on board, such as the AX5043 transceiver board described in this paper.

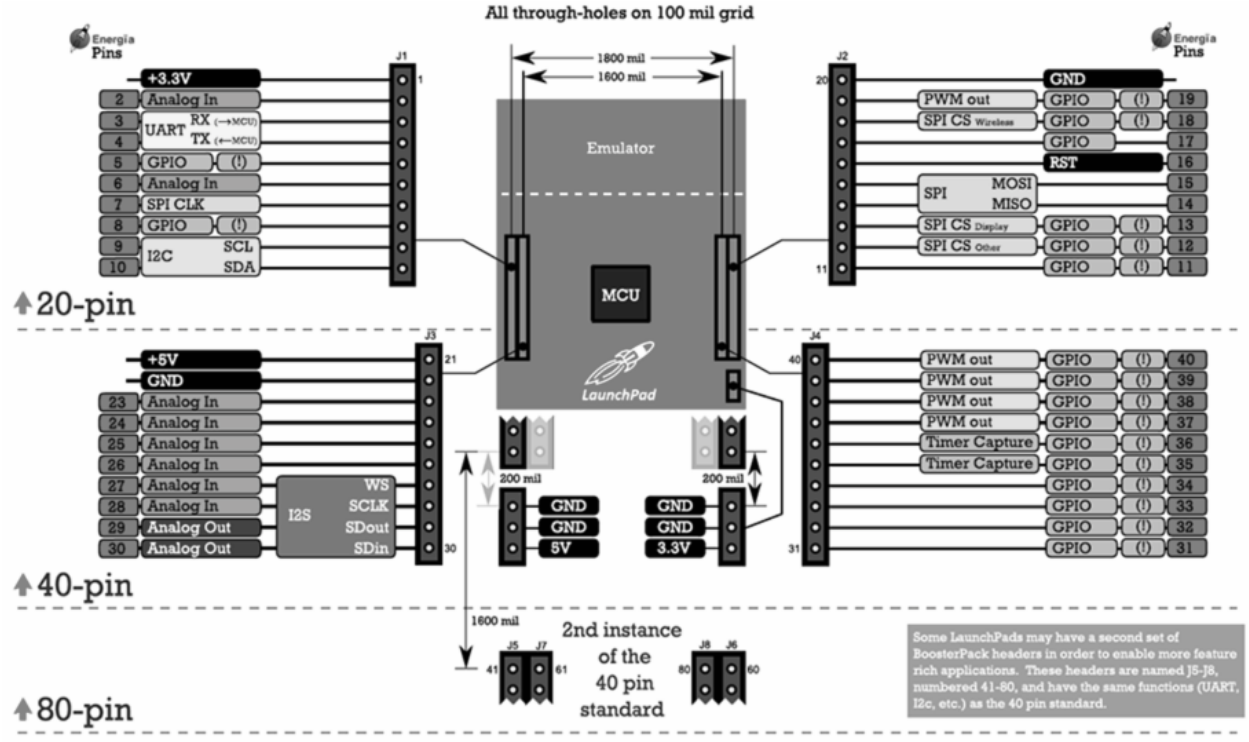



**FIGURE 13 - BOOSTERPACK™ MODULE PINOUT STANDARD**[xviii]

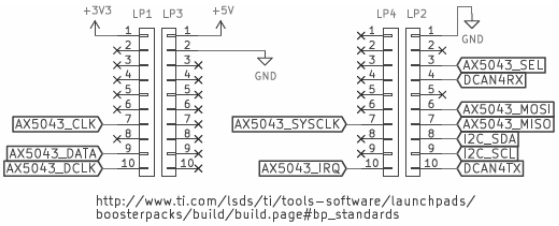



**FIGURE 14 HEADER SCHEMATIC ON AX5043 BOOSTERPACK™**

The AX5043 transceiver board includes headers for the 40-pin variant of the BoosterPack™ standard, using several pints for power and communication between the LaunchPad™ Development Kit and the AX5043 transceiver board.

| Pin # | Description | | Pin # | Description |
|---|---|---|---|---|
| (sorted by pin#) | | | (grouped by purpose) | |
| 1 | Power – +3.3V | | 1 | Power – +3.3V |
| 7 | SPI – CLK | | 20 | Power – Ground |
| 9 | I2C (1) – SCL | | 21 | Power – +5V |
| 10 | I2C (1) – SDA | | 22 | Power – Ground |
| 11 | CAN – TX | | 11 | SPI – CLK |
| 12 | I2C (2) – SCL | | 14 | SPI – MISO |
| 13 | I2C (2) – SDA | | 15 | SPI – MOSI |
| 14 | SPI – MISO | | 18 | SPI – EN |
| 15 | SPI – MOSI | | 9 | I2C (1) – SCL |
| 17 | CAN – RX | | 10 | I2C (1) – SDA |
| 18 | SPI – EN | | 12 | I2C (2) – SCL |
| 20 | Power – Ground | | 13 | I2C (2) – SDA |
| 21 | Power – +5V | | 11 | CAN – TX |
| 22 | Power – Ground | | 17 | CAN – RX |
| 31 | IRQ | | 31 | IRQ |
| 32 | Clock Output | | 32 | Clock Output |

## PCB

The AX5043 transceiver board is laid out as a 4-layer board:

- Front Copper Layer – RF signals are generally routed on the front copper layer.
- Inner Copper Layer #1 – Generally provides a ground layer separating the RF signals on the front layer from power lines and data signals.
- Inner Copper Layer #2 – Generally provides a +3.3V power layer.
- Back Copper Layer – Data signals are generally routed on the back copper layer.

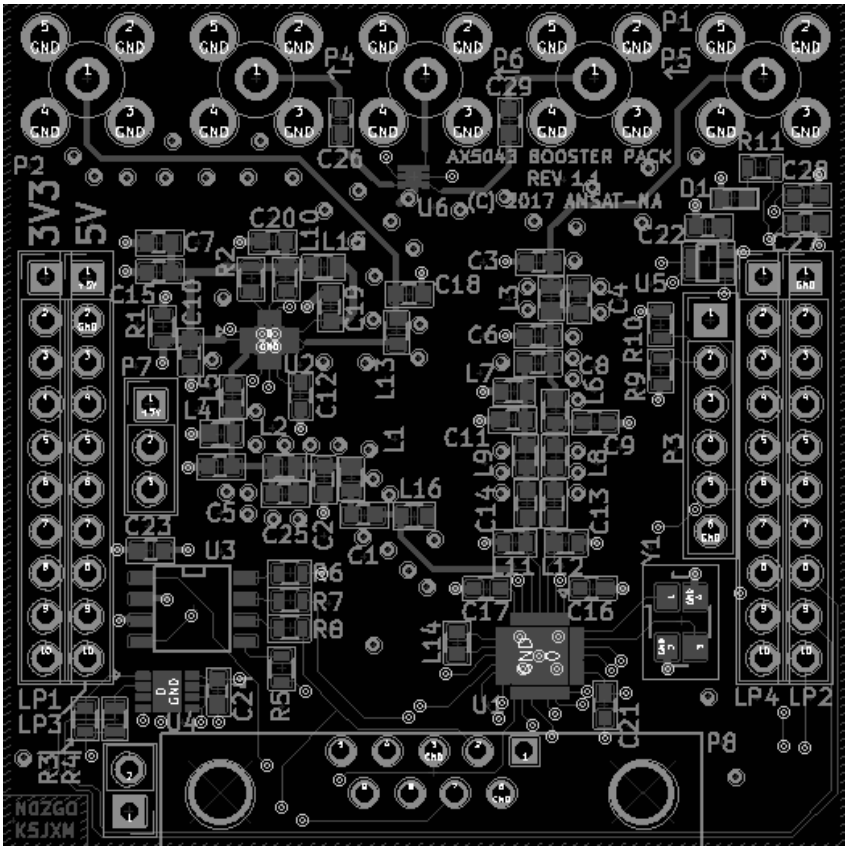



**FIGURE 15 - AX5043 PCB LAYOUT**

## TEST SOFTWARE

Test software was written to exercise the AX5043 BoosterPack™ as both a transmitter and a receiver. The software configures the AX5043 for GMSK[xix] with a symbol rate of 2400 baud. The transmitter obtains the temperature from the MAX31725 and transmits that value along with a call sign. The receiver demodulates that transmission and displays the temperature to the LaunchPad™ serial port.

# SUPERVISOR BOARD

## PROCESSOR

Previous satellites have used either expensive and hard-to-source radiation-hardened microprocessors or commercial microcontrollers without significant radiation tolerance. Neither of these solutions lends themselves well to a low-cost, long-lived satellite control IHU.

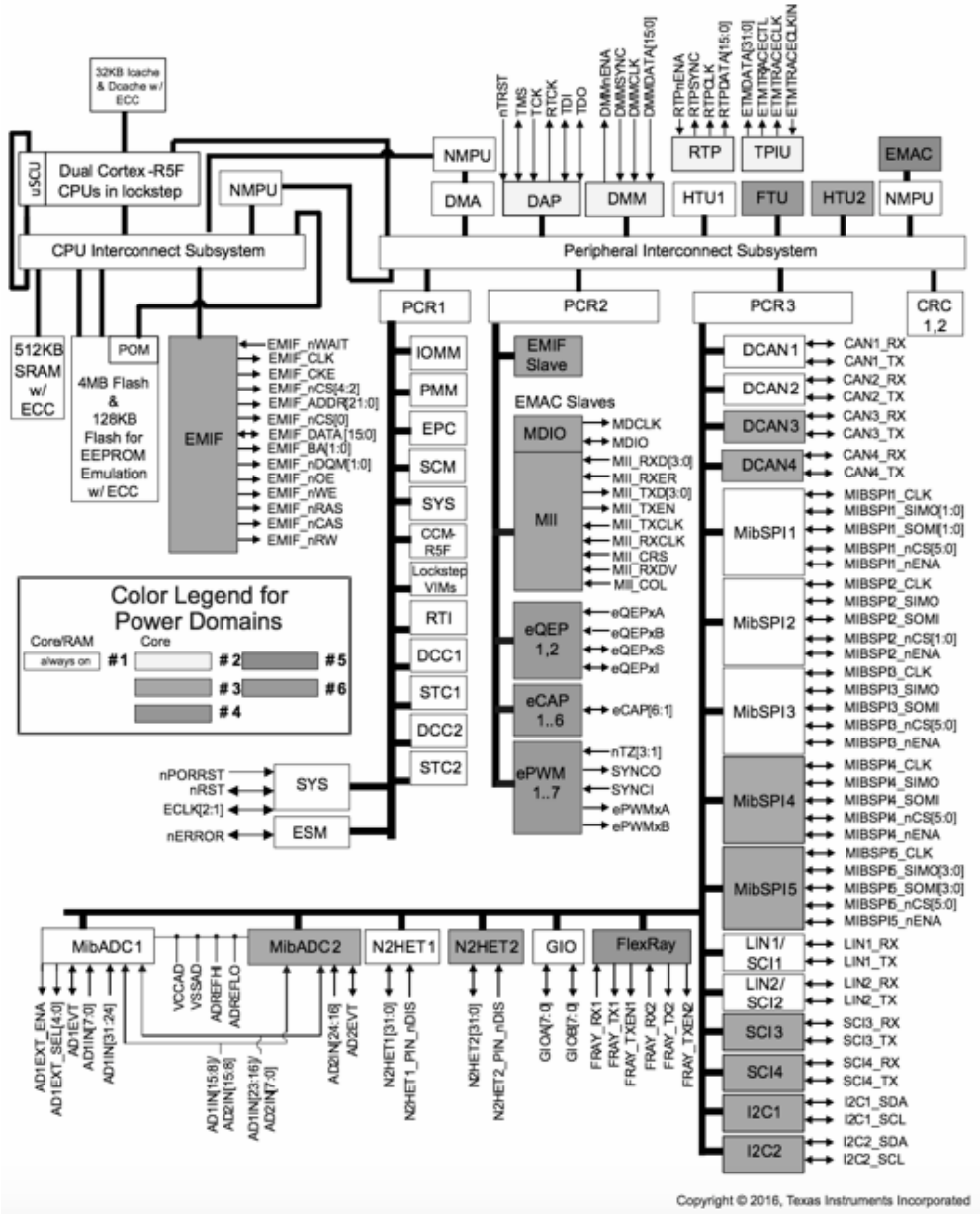



**FIGURE 16 - TMS570 HERCULES™ FUNCTIONAL BLOCK DIAGRAM[xx]**

Our IHU design incorporates the Texas Instruments TMS570 Hercules™ processor in a redundant primary/standby configuration. The TMS570 is targeted at automotive and other

safety-critical applications, such as vehicle stability system. In order to meet automotive reliability standards, each TMS570 microcontroller contains two lock-step ARM® Cortex™-R processors, EDAC data busses and memories, and a fault detection unit. A bit-flip caused by radiation or cosmic particle strike can be detected, corrected, and signaled to system software. In the rare case of an unrecoverable error, an active-low FAULT output from the IC provides an indication to other system components that the processor has failed. This feature is central to the primary/standby configuration of the IHU board, and will be described later in this document.

The specific device selected is the TMS570LS914, which has 1 MB of flash and 512 kB static RAM. As mentioned previously, these memories are protected against radiation and cosmic ray corruption by EDAC logic. Additionally, the TMS570 provides the necessary SPI, I2C, and CAN bus interfaces to the AX5043, MAX3051, and MAX31725 chips.

### Failover Concept

The IHU board contains two isolated supervisor systems, each with its own power, processing, and digital radio subsystems. One of these systems is designated as the primary IHU, while the other system is designated as the secondary IHU. Each IHU is fed from a separate power rail, for maximum fault tolerance. Ideally, one-half of the satellite power generation and storage will be routed to the primary IHU, while the other half will be routed to the secondary IHU. Each IHU is attached to a digital radio transceiver chip, a temperature sensor, and a CAN transceiver. An optically-isolated path, using a UART, is available for direct supervisor-to-supervisor communications.

A very simple digital logic circuit, called the arbiter, is used to examine the active-low FAULT outputs from each of the two isolated supervisor systems. Out of reset, the primary supervisor has control of all the IHU board I/O, while the secondary supervisor is in a low-power monitoring state. Should the FAULT signal from the primary supervisor go active, the arbiter logic routes all of the IHU board's I/O to the secondary supervisor. The secondary supervisor immediately takes command of all IHU I/O and begins to control spacecraft operation. This fail-over is a latching condition, and only occurs if the FAULT signal from the secondary supervisor is not active, as there is little gain in failing over to a failed supervisor. A power-cycle system is under consideration, which would detect that both FAULT outputs are active for some period of time and provide a short power-off, power-on sequence to both supervisors. It is important to note that power for the arbiter is drawn from either the primary or secondary supervisor power systems via a diode isolation arrangement. As the FAULT signal is active-low, failure of one power supply will automatically cause that FAULT signal to be asserted.

The secondary supervisor can transfer control back to the primary supervisor via a GPIO pin on the arbiter, but only if the FAULT output from the primary supervisor is not active.

All possible failure modes of the redundant system are still under investigation, and the exact mechanism of this system is subject to change.

# DEVELOPMENT TOOLS

## CODE COMPOSER STUDIO™

Code Composer Studio™ is an integrated development environment, provided by Texas Instruments, that supports Texas Instrument's microcontrollers. Specifically, Code Composer Studio™ supports the Texas Instruments TMS570 Hercules™ processor on the TMS570LC43x LaunchPad™ Development Kit. Code Composer Studio™ is available free of charge from the Texas Instruments web site[xxi].

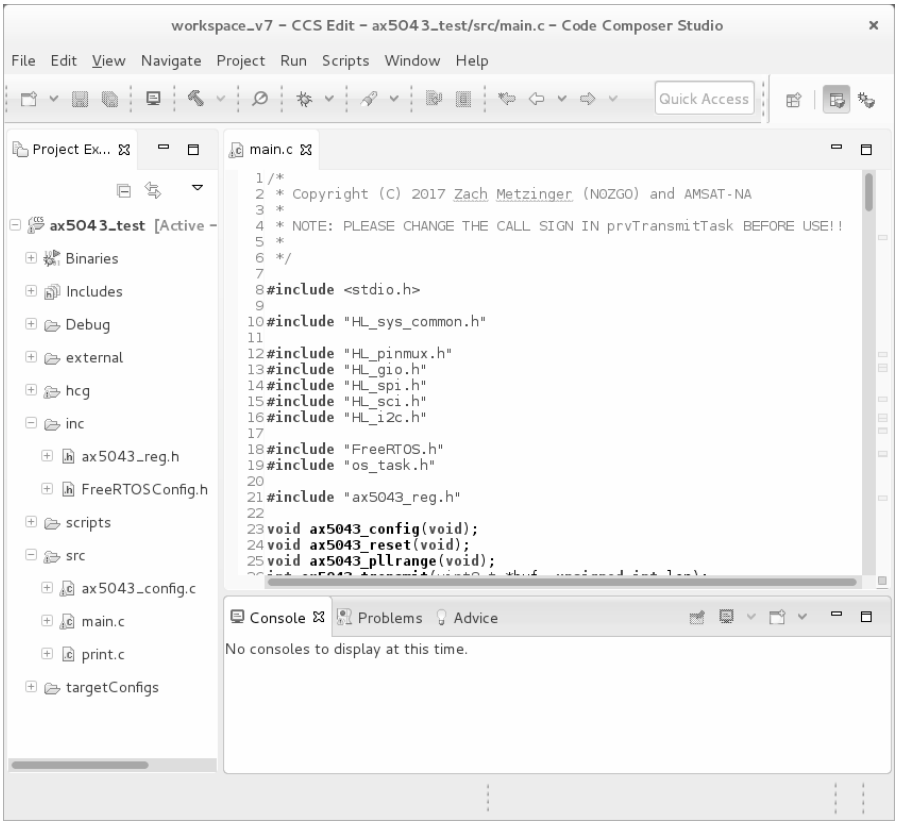



**FIGURE 17 - CODE COMPOSER STUDIO™**

## AX-RadioLAB

AX-RadioLAB for AX5043 is a software package from ON Semiconductor that generates source code to configure and operate the AX5043 transceiver[xxii]. The generated software includes register values to configure the AX5043 into the desired mode. The programmer can then tweak the register values according to the AX5043 programmer's guide. However, the programmer's guide is not comprehensive and some register settings require trial-and-error to obtain a working configuration. AX-RadioLab for AX5043 is available for download from the ON Semiconductor web site[xxiii].

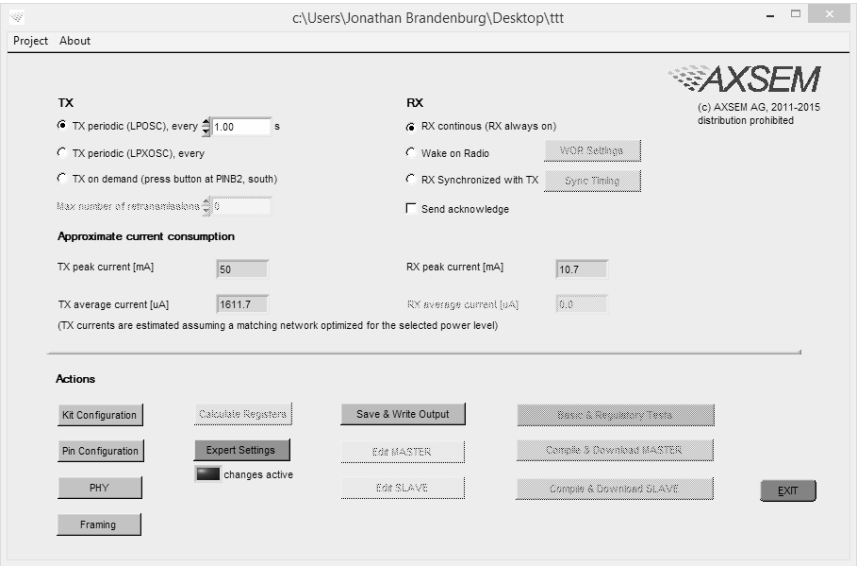



**FIGURE 18 - AX-RADIOLAB**

## FreeRTOS

The team is using FreeRTOS as a real-time operating system for the supervisor board's processor. FreeRTOS has been used on previous AMSAT missions and has been helpful in dividing up responsibility when writing the supervisor software. FreeRTOS is available free-of-charge from the project's web site[xxiv].

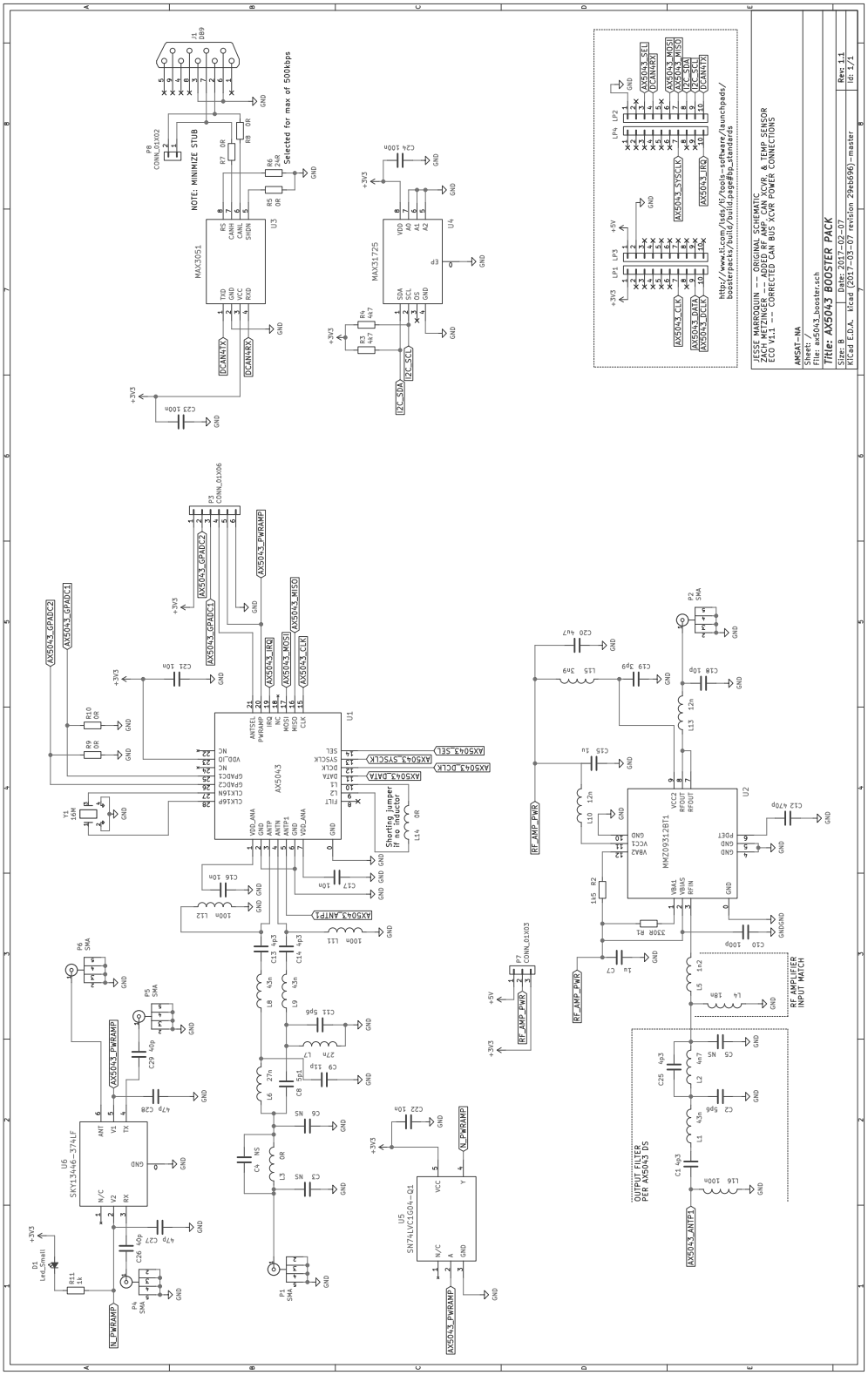
## KiCad

KiCad is an open source electronics design package allowing the team to create schematics and layout the PCBs[xxv]. KiCad is available free-of-charge.

## Next Steps

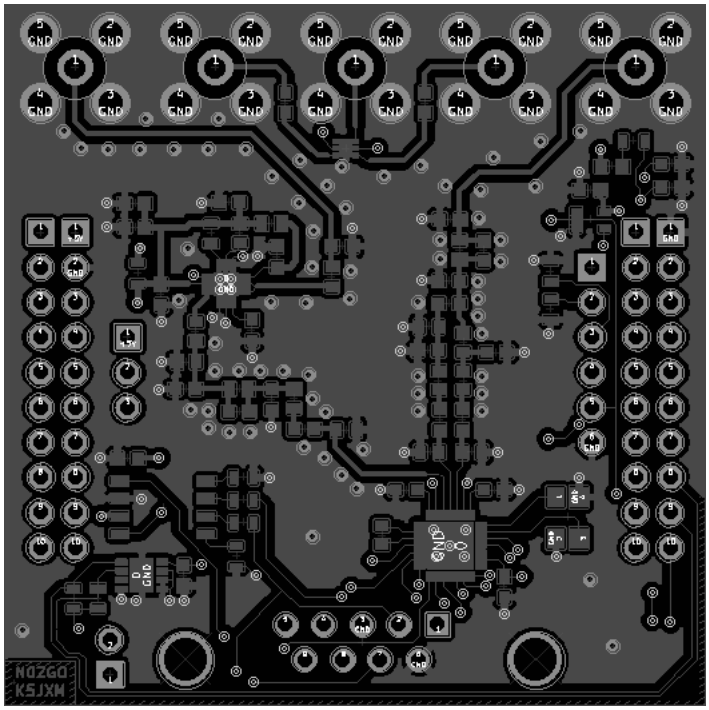
Design and layout of the Hercules™-based supervisor board, including the failover circuitry, is in progress with the design and development of the IHU software pending.

# APPENDIX – AX5043 BOOSTERPACK™ SCHEMATIC

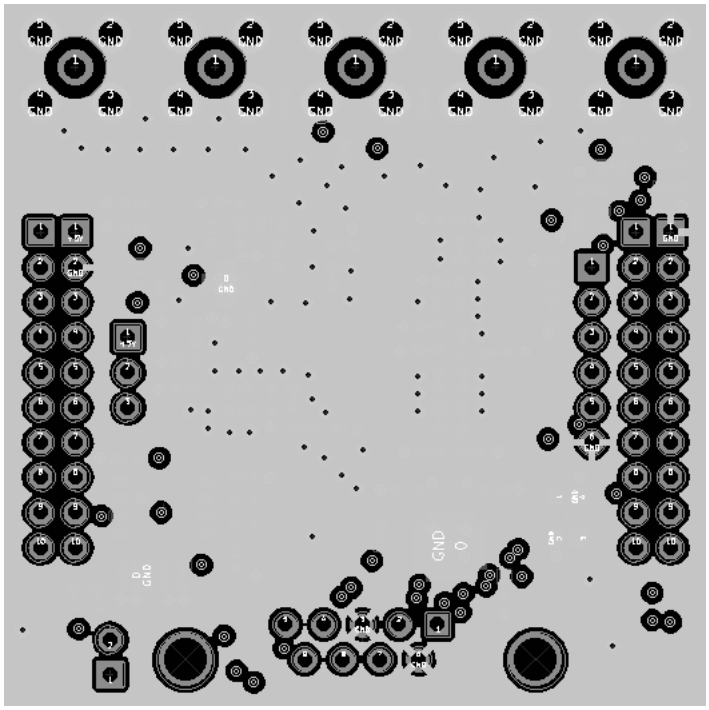


# APPENDIX – AX5043 LAYOUT

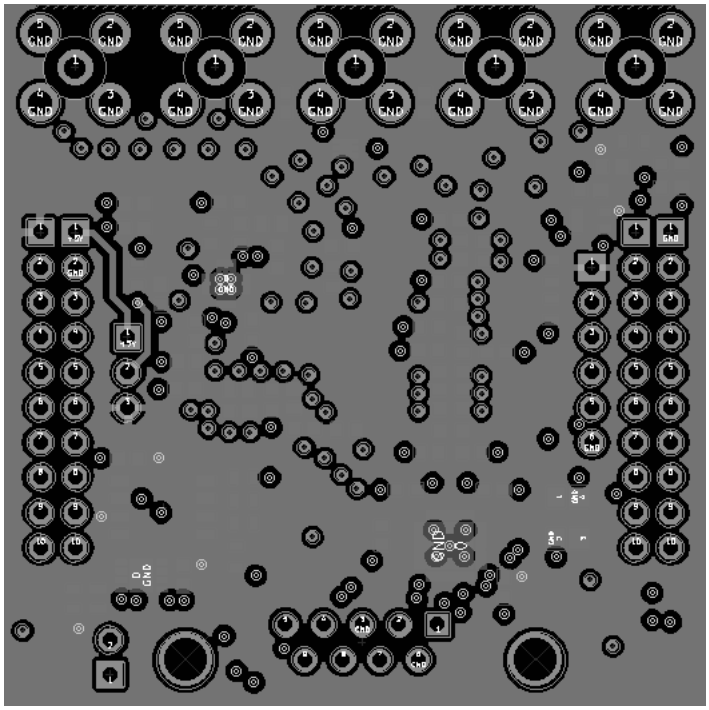
## FRONT COPPER LAYER



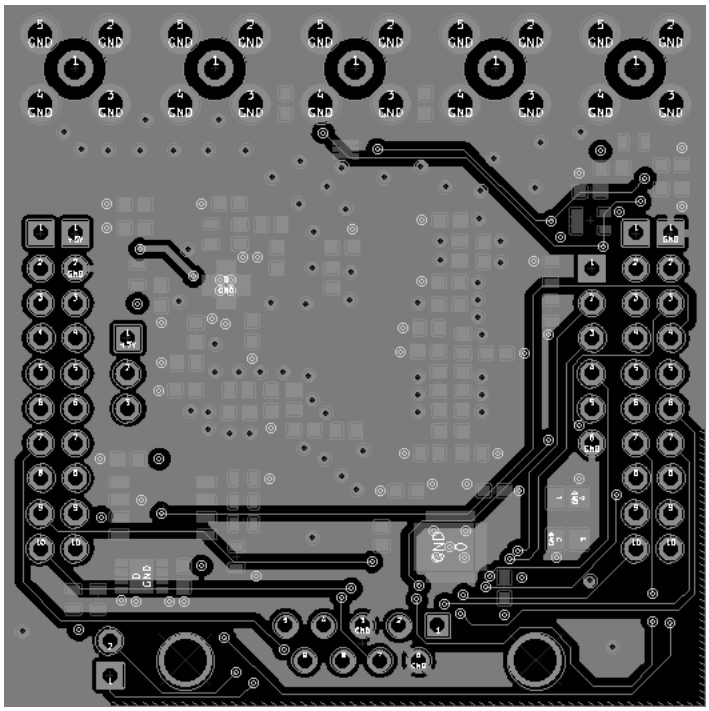

## INNER COPPER LAYER #1



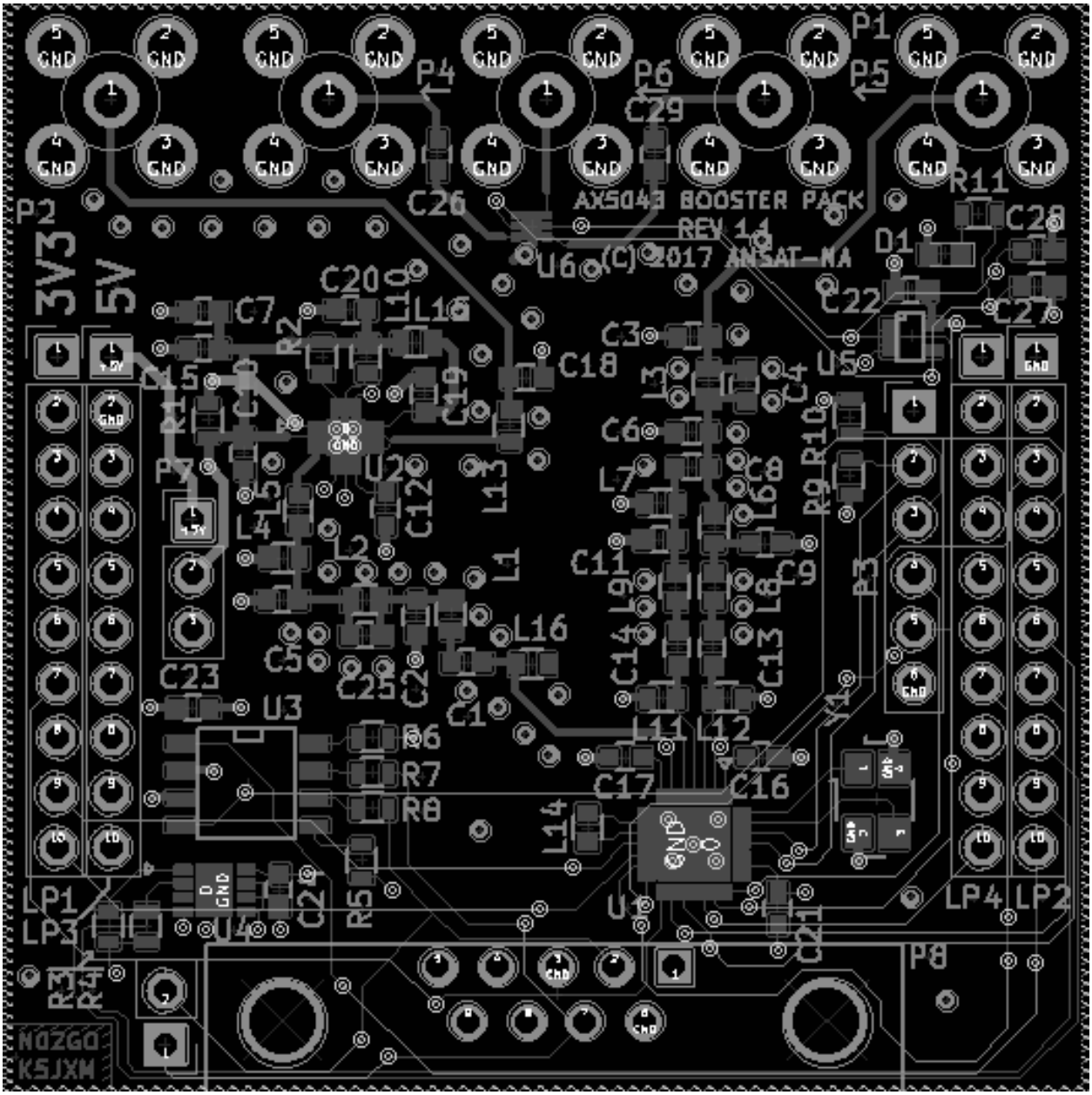
## INNER COPPER LAYER #2



## BACK COPPER LAYER



ALL COPPER LAYERS (WITHOUT FILL)



# APPENDIX – AX5043 TEST SOFTWARE

## MAIN.C

```
/*
 * Copyright (C) 2017 Zach Metzinger (N0ZGO) and AMSAT-NA
 *
 * NOTE: PLEASE CHANGE THE CALL SIGN IN prvTransmitTask BEFORE USE!!
 *
 */

#include <stdio.h>

#include "HL_sys_common.h"

#include "HL_pinmux.h"
#include "HL_gio.h"
#include "HL_spi.h"
#include "HL_sci.h"
#include "HL_i2c.h"

#include "FreeRTOS.h"
#include "os_task.h"

#include "ax5043_reg.h"

void ax5043_config(void);
void ax5043_reset(void);
void ax5043_pllrange(void);
int ax5043_transmit(uint8_t *buf, unsigned int len);

unsigned int ax5043_readreg(unsigned int reg);
void ax5043_writereg(unsigned int reg, unsigned int val);

void print(char *s);

#define MAX31725_ADDR (0x90 >> 1) /* 7-bit address mode */

/* If 1, this board will transmit. If zero, the board will receive. */
#define TRANSMITTER 1

int g_temp = 0;

static void prvBlinkTask(void *pvParameters)
{
  /* Block for 500ms. */
  const TickType_t xDelay = 500 / portTICK_PERIOD_MS;

  while (1) {
    gioToggleBit(gioPORTB, 6);
    vTaskDelay( xDelay );
  }
}

static void prvTempSensorTask(void *pbParameters)
{
  unsigned int i, j;
  char b[80];
  unsigned char buf[2];
  const TickType_t xDelay = 1000 / portTICK_PERIOD_MS;

  print("I2C init.\r\n");
  i2cInit();

  while (1) {
    /* Write device register address */
    //print("I2C Write register address\r\n");
    i2cSetSlaveAdd(i2cREG1, MAX31725_ADDR);
    i2cSetDirection(i2cREG1, I2C_TRANSMITTER);
    i2cSetCount(i2cREG1, 1);
    i2cSetMode(i2cREG1, I2C_MASTER);
    i2cSetStop(i2cREG1);
    i2cSetStart(i2cREG1);
    i2cSendByte(i2cREG1, 0x00);
    while(i2cIsBusBusy(i2cREG1) == true);
    while(i2cIsStopDetected(i2cREG1) == 0);
    i2cClearSCD(i2cREG1);

    /* Read data */
    //print("I2C Read data.\r\n");
    while(i2cIsMasterReady(i2cREG1) != true);
    i2cSetSlaveAdd(i2cREG1, MAX31725_ADDR);
    i2cSetDirection(i2cREG1, I2C_RECEIVER);
    i2cSetCount(i2cREG1, 2);
```

```
    i2cSetMode(i2cREG1, I2C_MASTER);
    i2cSetStop(i2cREG1);
    i2cSetStart(i2cREG1);
    i2cReceive(i2cREG1, 2, buf);
    while(i2cIsBusBusy(i2cREG1) == true);
    while(i2cIsStopDetected(i2cREG1) == 0);
    i2cClearSCD(i2cREG1);

    /* Convert to fractional degrees Fahrenheit */
    i = (buf[0] << 2) + (buf[1] >> 6);
    i *= 9 << 2;
    i /= 5 << 2;
    i += 32 << 2;
    j = (i & 0x3) * 25;
    i >>= 2;

    gioToggleBit(gioPORTB, 7);
    snprintf(b, 80, "Temperature %u.%02u (F)\r\n", i, j);
    g_temp = i;
    g_temp += (j > 50) ? 1 : 0;
    print(b);

    vTaskDelay( xDelay );
  }
}

static void prvTransmitTask(void *pbParameters)
{
  const TickType_t xDelay = 1100 / portTICK_PERIOD_MS;
  unsigned int altfreq = 0;
  uint8_t buf[12] = { 'N', '0', 'Z', 'G', 'O', ' ', 'T', 'E', 'S', 'T', ' ', 0x00 };

  ax5043_config();

  ax5043_pllrange();

  ax5043_synthtx();

  ax5043_setfreq(435302500);

  while (1) {
#if 0
    if (altfreq++) {
      ax5043_setfreq(435305000);
      altfreq = 0;
    } else {
      ax5043_setfreq(435300000);
    }
#endif
    gioToggleBit(gioPORTB, 6);
    buf[11] = g_temp; /* Transmit temperature */
    ax5043_transmit(buf, 12);
    vTaskDelay( xDelay );
  }
}

static void prvReceiveTask(void *pbParameters)
{
  volatile unsigned int x;
  uint8_t d, b[20];
  int i;

  ax5043_config();

  ax5043_pllrange();

  ax5043_receive();

  ax5043_setfreq(435302500);

  i = 0;

  while (1) {
    if (!(i++ % 10)) {
      snprintf(b, 20, "-> %u\r\n", i);
      print(b);
    }

    x = ax5043_readreg(AX_RADIOSTATE);
    x = ax5043_readreg(AX_PWRMODE);

    x = ax5043_readreg(AX_FIFOCOUNT1) << 8;
    x += ax5043_readreg(AX_FIFOCOUNT0);

    if (x) {
      while (x--) {
         gioToggleBit(gioPORTB, 7);
         d = ax5043_readreg(AX_FIFODATA);
```

```
          snprintf(b, 20, "0x%02x %c\r\n", d, ((d >= 0x20) && (d < 0x80)) ? d : '.');
          print(b);
          gioToggleBit(gioPORTB, 7);
        }
      } else {
        gioToggleBit(gioPORTB, 6);
      }
#if 0
      if (++i > 10) {
        i = 0;
        d = ax5043_readreg(AX_RSSI);
        snprintf(b, 20, "RSSI = 0x%02x\r\n", d);
        print(b);
      }

      if (sciIsRxReady(sciREG1)) {
        sciSendByte(sciREG1, sciReceiveByte(sciREG1));
      } else {
        vTaskDelay( 1500 / portTICK_PERIOD_MS );
      }
#endif
      vTaskDelay( 100 / portTICK_PERIOD_MS );
    }
}

int app_main(void)
{
  int i;

  gioInit();
  muxInit();
  sciInit();

  print("AX5043 demo\r\n");

  spiInit();

  print("Verifying AX5043 ID .. ");
  if (ax5043_readreg(0x00) != 0x51) {
    print("FAIL. Halted.\r\n");
    while (1);
  }
  print("OK!\r\n");

  print("Resetting AX5043\r\n");
  ax5043_reset();

  print("Verifying AX5043 ID .. ");
  if (ax5043_readreg(0x01) != 0xc5) {
    print("FAIL. Halted.\r\n");
    while (1);
  }
  print("OK!\r\n");

  //xTaskCreate(prvBlinkTask, "Blink", configMINIMAL_STACK_SIZE, NULL, (tskIDLE_PRIORITY + 1), NULL);
#if (TRANSMITTER != 0)
  xTaskCreate(prvTempSensorTask, "TempSensor", configMINIMAL_STACK_SIZE*2, NULL, (tskIDLE_PRIORITY + 1), NULL);
  xTaskCreate(prvTransmitTask, "Transmit", configMINIMAL_STACK_SIZE*2, NULL, (tskIDLE_PRIORITY + 2), NULL);
#else
  xTaskCreate(prvReceiveTask, "Receive", configMINIMAL_STACK_SIZE*2, NULL, (tskIDLE_PRIORITY + 2), NULL);
#endif


  /* Start the tasks and timer running. */
  vTaskStartScheduler();

  print("vTaskStartScheduler() failed!\r\n");
  while (1);
}
```

## AX5043_CONFIG.C

```
#include "HL_sys_common.h"
#include "HL_spi.h"

#include "ax5043_reg.h"

void ax5043_writereg(unsigned int reg, unsigned int val)
{
  spiDAT1_t dc = {.WDEL = false, .CS_HOLD = true, .DFSEL = SPI_FMT_0, .CSNR = 1 };
  uint16_t srcbuf[3];
  uint16_t dstbuf[3];

  srcbuf[0] = 0x00f0 | ((reg & 0xf00) >> 8);
  srcbuf[1] = (reg & 0xff);
  srcbuf[2] = val & 0xff;

  spiTransmitAndReceiveData(spiREG3, &dc, 3, srcbuf, dstbuf);
}

unsigned int ax5043_readreg(unsigned int reg)
{
  spiDAT1_t dc = {.WDEL = false, .CS_HOLD = true, .DFSEL = SPI_FMT_0, .CSNR = 1 };
  uint16_t srcbuf[3];
  uint16_t dstbuf[3];

  srcbuf[0] = 0x0070 | ((reg & 0xf00) >> 8);
  srcbuf[1] = (reg & 0xff);
  srcbuf[2] = 0x0000;

  spiTransmitAndReceiveData(spiREG3, &dc, 3, srcbuf, dstbuf);

  return (dstbuf[2] & 0xff);
}

void ax5043_reset(void)
{
  /* PWRMODE -- reset, then bring chip out of deep sleep */
  ax5043_writereg(AX_PWRMODE, AX_PWRMODE_RST);
  ax5043_writereg(AX_PWRMODE, 0x00);
  ax5043_writereg(AX_PWRMODE,
                  AX_PWRMODE_REFEN |
                  AX_PWRMODE_XOEN |
                  AX_PWRMODE_MODE_FIFOON);
}

void ax5043_pllrange(void)
{
  ax5043_writereg(AX_PLLRANGINGA, ax5043_readreg(AX_PLLRANGINGA) | 0x10);
  while (ax5043_readreg(AX_PLLRANGINGA) & 0x10);
  if (ax5043_readreg(AX_PLLRANGINGA) & 0x40) {
    while (1);
  }
}

void ax5043_synthtx(void)
{
  ax5043_writereg(AX_PWRMODE,
                  AX_PWRMODE_REFEN |
                  AX_PWRMODE_XOEN |
                  AX_PWRMODE_MODE_SYNTHTX);
}


void ax5043_fulltx(void)
{
  ax5043_writereg(AX_PWRMODE,
                  AX_PWRMODE_REFEN |
                  AX_PWRMODE_XOEN |
                  AX_PWRMODE_MODE_FULLTX);
}

void ax5043_synthrx(void)
{
  ax5043_writereg(AX_PWRMODE,
                  AX_PWRMODE_REFEN |
                  AX_PWRMODE_XOEN |
                  AX_PWRMODE_MODE_SYNTHRX);
}

void ax5043_fullrx(void)
{
  ax5043_writereg(AX_PWRMODE,
                  AX_PWRMODE_REFEN |
                  AX_PWRMODE_XOEN |
                  AX_PWRMODE_MODE_FULLRX);
```

```
}

int ax5043_transmit(uint8_t *buf, unsigned int len)
{
  int i;

  /* FIXME: For now, limit transmit size to 128 bytes of payload */
  if (len > 128) {
    return -1;
  }

  ax5043_fulltx();
  while ((ax5043_readreg(AX_POWSTAT) & 0x7f) != 0x7f);
  ax5043_writereg(AX_FIFOSTAT, 0x03); /* Clear FIFO and flags */
  ax5043_writereg(AX_FIFOSTAT, 0x00); /* NOP */
  i = ax5043_readreg(AX_FIFOSTAT);
  /* Preamble */
  ax5043_writereg(AX_FIFODATA, 0x62); /* REPEATDATA command */
  ax5043_writereg(AX_FIFODATA, 0x13); /* Transmit flags == RAW | PKTSTART | PKTEND */
  ax5043_writereg(AX_FIFODATA, 0x10); /* # of times to repeat data */
  ax5043_writereg(AX_FIFODATA, 0x11); /* data */
  /* Payload */
  ax5043_writereg(AX_FIFODATA, 0xE1); /* Transmit DATA command */
  ax5043_writereg(AX_FIFODATA, len+2); /* Includes transmit flags byte */
  ax5043_writereg(AX_FIFODATA, 0x03); /* Transmit flags == PKTSTART | PKTEND */
  ax5043_writereg(AX_FIFODATA, len);
  while (len--) {
    ax5043_writereg(AX_FIFODATA, *buf++);
  }
  i = ax5043_readreg(AX_FIFOSTAT);
  while (!(ax5043_readreg(AX_XTALSTATUS) & 0x01));
  ax5043_writereg(AX_PWRAMP, 0x01); /* Turn on power amplifier */
  ax5043_writereg(AX_FIFOSTAT, 0x04); /* Commit data and begin transmit */
  while (ax5043_readreg(AX_RADIOSTATE) & 0x0f); /* Wait for TX complete */
  ax5043_writereg(AX_PWRAMP, 0x00); /* Turn off power amplifier */
  ax5043_synthtx();

  return 0;
}

/*
 * Start the receiver. FIXME: Currently, all old packets are cleared.
 *
 */
int ax5043_receive(void)
{
  ax5043_synthrx();
  ax5043_writereg(AX_FIFOSTAT, 0x03); /* Clear FIFO and flags */
  ax5043_writereg(AX_FIFOSTAT, 0x00); /* NOP */
  ax5043_fullrx();
}

/*
 * Set frequency of AX5043. Unfortunately, 64-bit math is used here due to the large
 * scaling factors, but floating-point math might be just as bad if we went that direction.
 *
 * Remember to keep in mind the VCO auto-ranging requirements: In the 70cm band, frequency
 * changes larger than 2.5 MHz require a new auto-ranging to be performed.
 *
 */
int ax5043_setfreq(uint32_t hz)
{
  uint64_t w;

  /* Multiply, first, to preserve accuracy */
  w = hz;
  w <<= 24;

  /* Divide by crystal frequency, assumed to be 16 MHz */
  w /= 16000000;

  /* Set LSB, per AX5043 documentation, to prevent synthesizer spurs */
  w |= 1;

  /* Upper bits should be zero, otherwise error */
  if (w >> 32) {
    /* Do not change freq. */
    return -1;
  }

  /* Set frequency */
  ax5043_writereg(AX_FREQA3, (w >> 24) & 0xff);
  ax5043_writereg(AX_FREQA2, (w >> 16) & 0xff);
  ax5043_writereg(AX_FREQA1, (w >> 8) & 0xff);
  ax5043_writereg(AX_FREQA0, (w) & 0xff);

  return 0;
}
```

```
void ax5043_config(void)
{
  /* Radiolab-generated settings for FSK */
  ax5043_writereg(AX_MODULATION, 0x08);
  ax5043_writereg(AX_ENCODING, 0x07);
  ax5043_writereg(AX_FRAMING, 0x24);
  ax5043_writereg(AX_FEC, 0x00);
  ax5043_writereg(AX_PINFUNCSYSCLK, 0x01);
  ax5043_writereg(AX_PINFUNCDCLK, 0x01);
  ax5043_writereg(AX_PINFUNCDATA, 0x01);
  ax5043_writereg(AX_PINFUNCANTSEL, 0x01);
  ax5043_writereg(AX_PINFUNCPWRAMP, 0x06);
  ax5043_writereg(AX_PLLLOOP, 0x0b);
  ax5043_writereg(AX_PLLCPI, 0x10);
  ax5043_writereg(AX_PLLVCODIV, 0x24);
  ax5043_writereg(AX_FREQA3, 0x1b);
  ax5043_writereg(AX_FREQA2, 0x34);
  ax5043_writereg(AX_FREQA1, 0xcc);
  ax5043_writereg(AX_FREQA0, 0xcd);
  ax5043_writereg(AX_WAKEUPXOEARLY, 0x01);
  ax5043_writereg(AX_IFFREQ1, 0x01);
  ax5043_writereg(AX_IFFREQ0, 0x06);
  ax5043_writereg(AX_DECIMATION, 0x37);
  ax5043_writereg(AX_RXDATARATE2, 0x00);
  ax5043_writereg(AX_RXDATARATE1, 0x3C);
  ax5043_writereg(AX_RXDATARATE0, 0x9B);
  ax5043_writereg(AX_MAXDROFFSET2, 0x00);
  ax5043_writereg(AX_MAXDROFFSET1, 0x00);
  ax5043_writereg(AX_MAXDROFFSET0, 0x00);
  ax5043_writereg(AX_MAXRFOFFSET2, 0x80);
  ax5043_writereg(AX_MAXRFOFFSET1, 0x03);
  ax5043_writereg(AX_MAXRFOFFSET0, 0x00);
  ax5043_writereg(AX_FSKDMAX1, 0x00);
  ax5043_writereg(AX_FSKDMAX0, 0xA6);
  ax5043_writereg(AX_FSKDMIN1, 0xFF);
  ax5043_writereg(AX_FSKDMIN0, 0x5A);
  ax5043_writereg(AX_AMPLFILTER, 0x00);
  ax5043_writereg(AX_RXPARAMSETS, 0xF4);
  ax5043_writereg(AX_AGCGAIN0, 0xD7);
  ax5043_writereg(AX_AGCTARGET0, 0x84);
  ax5043_writereg(AX_TIMEGAIN0, 0xF8);
  ax5043_writereg(AX_DRGAIN0, 0xF2);
  ax5043_writereg(AX_PHASEGAIN0, 0xC3);
  ax5043_writereg(AX_FREQGAINA0, 0x0F);
  ax5043_writereg(AX_FREQGAINB0, 0x1F);
  ax5043_writereg(AX_FREQGAINC0, 0x0B);
  ax5043_writereg(AX_FREQGAIND0, 0x0B);
  ax5043_writereg(AX_AMPLGAIN0, 0x06);
  ax5043_writereg(AX_FREQDEV10, 0x00);
  ax5043_writereg(AX_FREQDEV00, 0x00);
  ax5043_writereg(AX_BBOFFSRES0, 0x00);
  ax5043_writereg(AX_AGCGAIN1, 0xD7);
  ax5043_writereg(AX_AGCTARGET1, 0x84);
  ax5043_writereg(AX_AGCAHYST1, 0x00);
  ax5043_writereg(AX_AGCMINMAX1, 0x00);
  ax5043_writereg(AX_TIMEGAIN1, 0xF6);
  ax5043_writereg(AX_DRGAIN1, 0xF1);
  ax5043_writereg(AX_PHASEGAIN1, 0xC3);
  ax5043_writereg(AX_FREQGAINA1, 0x0F);
  ax5043_writereg(AX_FREQGAINB1, 0x1F);
  ax5043_writereg(AX_FREQGAINC1, 0x0B);
  ax5043_writereg(AX_FREQGAIND1, 0x0B);
  ax5043_writereg(AX_AMPLGAIN1, 0x06);
  ax5043_writereg(AX_FREQDEV11, 0x00);
  ax5043_writereg(AX_FREQDEV01, 0x25);
  ax5043_writereg(AX_FOURFSK1, 0x16);
  ax5043_writereg(AX_BBOFFSRES1, 0x00);
  ax5043_writereg(AX_AGCGAIN3, 0xFF);
  ax5043_writereg(AX_AGCTARGET3, 0x84);
  ax5043_writereg(AX_AGCAHYST3, 0x00);
  ax5043_writereg(AX_AGCMINMAX3, 0x00);
  ax5043_writereg(AX_TIMEGAIN3, 0xF5);
  ax5043_writereg(AX_DRGAIN3, 0xF0);
  ax5043_writereg(AX_PHASEGAIN3, 0xC3);
  ax5043_writereg(AX_FREQGAINA3, 0x0F);
  ax5043_writereg(AX_FREQGAINB3, 0x1F);
  ax5043_writereg(AX_FREQGAINC3, 0x0D);
  ax5043_writereg(AX_FREQGAIND3, 0x0D);
  ax5043_writereg(AX_AMPLGAIN3, 0x06);
  ax5043_writereg(AX_FREQDEV13, 0x00);
  ax5043_writereg(AX_FREQDEV03, 0x25);
  ax5043_writereg(AX_FOURFSK3, 0x16);
  ax5043_writereg(AX_BBOFFSRES3, 0x00);
  ax5043_writereg(AX_MODCFGF, 0x02);
  ax5043_writereg(AX_FSKDEV2, 0x00);
  ax5043_writereg(AX_FSKDEV1, 0x02);
  ax5043_writereg(AX_FSKDEV0, 0x75);
```

```
  ax5043_writereg(AX_MODCFGA, 0x05); /* Use differential output */
  //ax5043_writereg(AX_MODCFGA, 0x06); /* Enable single-ended output to PA */
  ax5043_writereg(AX_TXRATE2, 0x00);
  ax5043_writereg(AX_TXRATE1, 0x09);
  ax5043_writereg(AX_TXRATE0, 0xD5);
  ax5043_writereg(AX_TXPWRCOEFFB1, 0x07);
  ax5043_writereg(AX_TXPWRCOEFFB0, 0xFF);
  ax5043_writereg(AX_PLLVCOI, 0x99);
  ax5043_writereg(AX_PLLRNGCLK, 0x03);
  ax5043_writereg(AX_XTALCAP, 0x03);
  ax5043_writereg(AX_BBTUNE, 0x0F);
  ax5043_writereg(AX_BBOFFSCAP, 0x77);
  ax5043_writereg(AX_PKTADDRCFG, 0x00);
  ax5043_writereg(AX_PKTLENCFG, 0xF0);
  ax5043_writereg(AX_PKTLENOFFSET, 0x06);
  ax5043_writereg(AX_PKTMAXLEN, 0xC8);
  ax5043_writereg(AX_MATCH0PAT3, 0xAA);
  ax5043_writereg(AX_MATCH0PAT2, 0xCC);
  ax5043_writereg(AX_MATCH0PAT1, 0xAA);
  ax5043_writereg(AX_MATCH0PAT0, 0xCC);
  ax5043_writereg(AX_MATCH0LEN, 0x00);
  ax5043_writereg(AX_MATCH1PAT1, 0x7E);
  ax5043_writereg(AX_MATCH1PAT0, 0x7E);
  ax5043_writereg(AX_MATCH1LEN, 0x87); /* We send RAW 0x7E during transmit. Logical that we need to look for it. */
  ax5043_writereg(AX_MATCH1MAX, 0x07);
  ax5043_writereg(AX_TMGTXBOOST, 0x32);
  ax5043_writereg(AX_TMGTXSETTLE, 0x14);
  ax5043_writereg(AX_TMGRXBOOST, 0x32);
  ax5043_writereg(AX_TMGRXSETTLE, 0x14);
  ax5043_writereg(AX_TMGRXOFFSACQ, 0x00);
  ax5043_writereg(AX_TMGRXCOARSEAGC, 0x73);
  ax5043_writereg(AX_TMGRXRSSI, 0x03);
  ax5043_writereg(AX_TMGRXPREAMBLE2, 0x17);
  ax5043_writereg(AX_RSSIABSTHR, 0xA0);
  ax5043_writereg(AX_BGNDRSSITHR, 0x00);
  ax5043_writereg(AX_PKTCHUNKSIZE, 0x0D);
  ax5043_writereg(AX_PKTMISCFLAGS, 0x04);
  ax5043_writereg(AX_PKTSTOREFLAGS, 0x7f);
  ax5043_writereg(AX_PKTACCEPTFLAGS, 0x30);
  ax5043_writereg(AX_DACVALUE1, 0x00);
  ax5043_writereg(AX_DACVALUE0, 0x00);
  ax5043_writereg(AX_DACCONFIG, 0x00);
  ax5043_writereg(0xF00, 0x0f);
  ax5043_writereg(0xF0D, 0x03);
  ax5043_writereg(0xF10, 0x03); /* 0x04 for TCXO < 43 MHz , 0x03 otherwise */
  ax5043_writereg(0xF11, 0x07); /* 0x07 for crystal, 0x00 for TCXO */
  ax5043_writereg(0xF18, 0x06);
  ax5043_writereg(0xF1C, 0x07);
  ax5043_writereg(0xF21, 0x68);
  ax5043_writereg(0xF22, 0xFF);
  ax5043_writereg(0xF23, 0x84);
  ax5043_writereg(0xF26, 0x98);
  ax5043_writereg(0xF34, 0x28);
  ax5043_writereg(0xF35, 0x10);
  ax5043_writereg(0xF44, 0x25);
}
```

## print.c

```
#include "HL_sys_common.h"
#include "HL_sci.h"

void print(char *s) {
  while (*s) {
    while ((sciREG1->FLR & 0x4) == 4);
    sciSendByte(sciREG1, *s++); /* send out text   */
  }
}
```

## AX5043_REG.H

```
#define AX_REVISION 0x0000 /*R R 01010001 SILICONREV(7:0) SiliconRevision */
#define AX_SCRATCH 0x0001 /*RW R 11000101 SCRATCH(7:0) ScratchRegister */
#define AX_PWRMODE 0x0002 /*RW R 011-0000 RST XOEN REFEN WDS PWRMODE(3:0) PowerMode VoltageRegulator */
#define AX_POWSTAT 0x0003 /*R R -------- SSUM SREF SVREF SVANA SVMODEM SBEVANA SBEVMODEM SVIO PowerManagementStatus */
#define AX_POWSTICKYSTAT 0x0004 /*R R -------- SSSUM SREF SSVREF SSVANA SSVMODEM SSBEVANA SSBEVMODEM SSVIO
        PowerManagementStickyStatus */
#define AX_POWIRQMASK 0x0005 /*RW R 00000000 MPWRGOOD MSREF MSVREF MSVANA MSVMODEM MSBEVANA MSBEVMODEM MSVIO
        PowerManagementInterruptMask */
#define AX_IRQMASK1 0x0006 /*RW R --000000 - - IRQMASK(13:8) IRQMask */
#define AX_IRQMASK0 0x0007 /*RW R 00000000 IRQMASK(7:0) IRQMask */
#define AX_RADIOEVENTMASK1 0x0008 /*RW R -------0 - - - - - - - RADIOEVENTMASK(8) RadioEventMask */
#define AX_RADIOEVENTMASK0 0x0009 /*RW R 00000000 RADIOEVENTMASK(7:0) RadioEventMask */
#define AX_IRQINVERSION1 0x000a /*RW R --000000 - - IRQINVERSION(13:8) IRQInversion */
#define AX_IRQINVERSION0 0x000b /*RW R 00000000 IRQINVERSION(7:0) IRQInversion */
#define AX_IRQREQUEST1 0x000c /*R R -------- - - IRQREQUEST(13:8) IRQRequest */
#define AX_IRQREQUEST0 0x000d /*R R -------- IRQREQUEST(7:0) IRQRequest */
#define AX_RADIOEVENTREQ1 0x000e /*R -------- - - - - - - - RADIOEVENTREQ(8) RadioEventRequest */
#define AX_RADIOEVENTREQ0 0x000f /*R -------- RADIOEVENTREQ(7:0) RadioEventRequest */
#define AX_MODULATION 0x0010 /*RW R ---01000 - - - RXHALFSPEED MODULATION(3:0) Modulation */
#define AX_ENCODING 0x0011 /*RW R ---00010 - - - ENCNOSYNC ENCMANCH ENCSCRAM ENCDIFF ENCINV Encoder/DecoderSettings */
#define AX_FRAMING 0x0012 /*RW R -0000000 FRMRX CRCMODE(2:0) FRMMODE(2:0) FABORT Framingsettings */
#define AX_CRCINIT3 0x0014 /*RW R 11111111 CRCINIT(31:24) CRCInitialisationData */
#define AX_CRCINIT2 0x0015 /*RW R 11111111 CRCINIT(23:16) CRCInitialisationData */
#define AX_CRCINIT1 0x0016 /*RW R 11111111 CRCINIT(15:8) CRCInitialisationData */
#define AX_CRCINIT0 0x0017 /*RW R 11111111 CRCINIT(7:0) CRCInitialisationData */
#define AX_FEC 0x0018 /*R 00000000 SHORTMEM RSTVITERBI FECNEG FECPOS FECINPSHIFT(2:0) FECENA FEC(Viterbi)Configuration
        */
#define AX_FECSYNC 0x0019 /*RW R 01100010 FECSYNC(7:0) InterleaverSynchronisationThreshold */
#define AX_FECSTATUS 0x001a /*R R -------- FECINV MAXMETRIC(6:0) FECStatus www.onsemi.com 26 Table28.CONTROLREGISTERMAP
        Synthesizer AX5043 Add Name Dir Ret Reset Bit Description 7 6 5 4 3 2 1 0 Status PinConfiguration */
#define AX_RADIOSTATE 0x001c /*R - ----0000 - - - - RADIOSTATE(3:0) RadioControllerState */
#define AX_XTALSTATUS 0x001d /*R R -------- - - - - - - - XTALRUN CrystalOscillatorStatus */
#define AX_PINSTATE 0x0020 /*R R -------- - - PSPWRAMP PSANTSEL PSIRQ PSDATA PSDCLK PSSYSCLK Pinstate */
#define AX_PINFUNCSYSCLK 0x0021 /*RW R 0--01000 PUSYSCLK - - PFSYSCLK(4:0) SYSCLKPinFunction */
#define AX_PINFUNCDCLK 0x0022 /*RW R 00---100 PUDCLK PIDCLK - - - PFDCLK(2:0) DCLKPinFunction */
#define AX_PINFUNCDATA 0x0023 /*RW R 10---111 PUDATA PIDATA - - - PFDATA(2:0) DATAPinFunction */
#define AX_PINFUNCIRQ 0x0024 /*RW R 00---011 PUIRQ PIIRQ - - - PFIRQ(2:0) IRQPinFunction */
#define AX_PINFUNCANTSEL 0x0025 /*RW R 00---110 PUANTSEL PIANTSEL - - - PFANTSEL(2:0) ANTSELPinFunction */
#define AX_PINFUNCPWRAMP 0x0026 /*RW R 00--0110 PUPWRAMP PIPWRAMP - - PFPWRAMP(3:0) PWRAMPPinFunction */
#define AX_PWRAMP 0x0027 /*RW R -------0 - - - - - - - PWRAMP PWRAMPControl FIFO */
#define AX_FIFOSTAT 0x0028 /*R R 0------- FIFOAUTOCOMMIT - FIFOFREETHR FIFOCNTTHR FIFOOVER FIFOUNDER FIFOFULL FIFOEMPTY
        FIFOControl W R FIFOCMD(5:0) */
#define AX_FIFODATA 0x0029 /*RW -------- FIFODATA(7:0) FIFOData */
#define AX_FIFOCOUNT1 0x002a /*R R -------0 - - - - - - - FIFOCOUNT(8) NumberofWordscurrentlyinFIFO */
#define AX_FIFOCOUNT0 0x002b /*R R 00000000 FIFOCOUNT(7:0) NumberofWordscurrentlyinFIFO */
#define AX_FIFOFREE1 0x002c /*R R -------1 - - - - - - - FIFOFREE(8) NumberofWordsthatcanbewrittentoFIFO */
#define AX_FIFOFREE0 0x002d /*R R 00000000 FIFOFREE(7:0) NumberofWordsthatcanbewrittentoFIFO */
#define AX_FIFOTHRESH1 0x002e /*RW R -------0 - - - - - - - FIFOTHRESH(8) FIFOThreshold */
#define AX_FIFOTHRESH0 0x002f /*RW R 00000000 FIFOTHRESH(7:0) FIFOThreshold */
#define AX_PLLLOOP 0x0030 /*RW R 0---1001 FREQB - - - DIRECT FILTEN FLT(1:0) PLLLoopFilterSettings */
#define AX_PLLCPI 0x0031 /*RW R 00001000 PLLCPI PLLChargePumpCurrent(Boosted) */
#define AX_PLLVCODIV 0x0032 /*RW R -000-000 - VCOIMAN VCO2INT VCOSEL - RFDIV REFDIV(1:0) PLLDividerSettings */
#define AX_PLLRANGINGA 0x0033 /*RW R 00001000 STICKYLOCK PLLLOCK RNGERR RNGSTART VCORA(3:0) PLLAutoranging */
#define AX_FREQA3 0x0034 /*RW R 00111001 FREQA(31:24) SynthesizerFrequency */
#define AX_FREQA2 0x0035 /*RW R 00110100 FREQA(23:16) SynthesizerFrequency */
#define AX_FREQA1 0x0036 /*RW R 11001100 FREQA(15:8) SynthesizerFrequency */
#define AX_FREQA0 0x0037 /*RW R 11001101 FREQA(7:0) SynthesizerFrequency */
#define AX_PLLLOOPBOOST 0x0038 /*RW R 0---1011 FREQB - - - DIRECT FILTEN FLT(1:0) PLLLoopFilterSettings(Boosted) */
#define AX_PLLCPIBOOST 0x0039 /*RW R 11001000 PLLCPI PLLChargePumpCurrent */
#define AX_PLLRANGINGB 0x003b /*RW R 00001000 STICKYLOCK PLLLOCK RNGERR RNGSTART VCORB(3:0) PLLAutoranging */
#define AX_FREQB3 0x003c /*RW R 00111001 FREQB(31:24) SynthesizerFrequency */
#define AX_FREQB2 0x003d /*RW R 00110100 FREQB(23:16) SynthesizerFrequency */
#define AX_FREQB1 0x003e /*RW R 11001100 FREQB(15:8) SynthesizerFrequency */
#define AX_FREQB0 0x003f /*RW R 11001101 FREQB(7:0) SynthesizerFrequency SignalStrength */
#define AX_RSSI 0x0040 /*R R -------- RSSI(7:0) ReceivedSignalStrengthIndicator */
#define AX_BGNDRSSI 0x0041 /*RW R 00000000 BGNDRSSI(7:0) BackgroundRSSI */
#define AX_DIVERSITY 0x0042 /*RW R ------00 - - - - - - ANTSEL DIVENA AntennaDiversityConfiguration www.onsemi.com 27
        Table28.CONTROLREGISTERMAP ReceiverTracking AX5043 Add Name Dir Ret Reset Bit Description 7 6 5 4 3 2 1 0 */
#define AX_AGCCOUNTER 0x0043 /*RW R -------- AGCCOUNTER(7:0) AGCCurrentValue */
#define AX_TRKDATARATE2 0x0045 /*R R -------- TRKDATARATE(23:16) DatarateTracking */
#define AX_TRKDATARATE1 0x0046 /*R R -------- TRKDATARATE(15:8) DatarateTracking */
#define AX_TRKDATARATE0 0x0047 /*R R -------- TRKDATARATE(7:0) DatarateTracking */
#define AX_TRKAMPL1 0x0048 /*R R -------- TRKAMPL(15:8) AmplitudeTracking */
#define AX_TRKAMPL0 0x0049 /*R R -------- TRKAMPL(7:0) AmplitudeTracking */
#define AX_TRKPHASE1 0x004a /*R R -------- - - - - TRKPHASE(11:8) PhaseTracking */
#define AX_TRKPHASE0 0x004b /*R R -------- TRKPHASE(7:0) PhaseTracking */
#define AX_TRKRFFREQ2 0x004d /*RW R -------- - - - - TRRFKFREQ(19:16) RFFrequencyTracking */
#define AX_TRKRFFREQ1 0x004e /*RW R -------- TRRFKFREQ(15:8) RFFrequencyTracking */
#define AX_TRKRFFREQ0 0x004f /*RW R -------- TRRFKFREQ(7:0) RFFrequencyTracking */
#define AX_TRKFREQ1 0x0050 /*RW R -------- TRKFREQ(15:8) FrequencyTracking */
#define AX_TRKFREQ0 0x0051 /*RW R -------- TRKFREQ(7:0) FrequencyTracking */
#define AX_TRKFSKDEMOD1 0x0052 /*R R -------- - - TRKFSKDEMOD(13:8) FSKDemodulatorTracking */
#define AX_TRKFSKDEMOD0 0x0053 /*R R -------- TRKFSKDEMOD(7:0) FSKDemodulatorTracking */
```

```
#define AX_TRKAFSKDEMOD1 0x0054 /*R R -------- TRKAFSKDEMOD(15:8) AFSKDemodulatorTracking */
#define AX_TRKAFSKDEMOD0 0x0055 /*R R -------- TRKAFSKDEMOD(7:0) AFSKDemodulatorTracking Timer WakeupTimer */
#define AX_TIMER2 0x0059 /*R - -------- TIMER(23:16) 1MHzTimer */
#define AX_TIMER1 0x005a /*R - -------- TIMER(15:8) 1MHzTimer */
#define AX_TIMER0 0x005b /*R - -------- TIMER(7:0) 1MHzTimer */
#define AX_WAKEUPTIMER1 0x0068 /*R R -------- WAKEUPTIMER(15:8) WakeupTimer */
#define AX_WAKEUPTIMER0 0x0069 /*R R -------- WAKEUPTIMER(7:0) WakeupTimer */
#define AX_WAKEUP1 0x006a /*RW R 00000000 WAKEUP(15:8) WakeupTime */
#define AX_WAKEUP0 0x006b /*RW R 00000000 WAKEUP(7:0) WakeupTime */
#define AX_WAKEUPFREQ1 0x006c /*RW R 00000000 WAKEUPFREQ(15:8) WakeupFrequency */
#define AX_WAKEUPFREQ0 0x006d /*RW R 00000000 WAKEUPFREQ(7:0) WakeupFrequency */
#define AX_WAKEUPXOEARLY 0x006e /*RW R 00000000 WAKEUPXOEARLY WakeupCrystalOscillatorEarly
        PhysicalLayerParametersReceiverParameters */
#define AX_IFFREQ1 0x0100 /*RW R 00010001 IFFREQ(15:8) 2ndLO/IFFrequency */
#define AX_IFFREQ0 0x0101 /*RW R 00100111 IFFREQ(7:0) 2ndLO/IFFrequency */
#define AX_DECIMATION 0x0102 /*RW R -0001101 - DECIMATION(6:0) DecimationFactor */
#define AX_RXDATARATE2 0x0103 /*RW R 00000000 RXDATARATE(23:16) ReceiverDatarate */
#define AX_RXDATARATE1 0x0104 /*RW R 00111101 RXDATARATE(15:8) ReceiverDatarate */
#define AX_RXDATARATE0 0x0105 /*RW R 10001010 RXDATARATE(7:0) ReceiverDatarate */
#define AX_MAXDROFFSET2 0x0106 /*RW R 00000000 MAXDROFFSET(23:16) MaximumReceiverDatarateOffset */
#define AX_MAXDROFFSET1 0x0107 /*RW R 00000000 MAXDROFFSET(15:8) MaximumReceiverDatarateOffset */
#define AX_MAXDROFFSET0 0x0108 /*RW R 10011110 MAXDROFFSET(7:0) MaximumReceiverDatarateOffset */
#define AX_MAXRFOFFSET2 0x0109 /*RW R 0---0000 FREQOFFSCORR - - - MAXRFOFFSET(19:16) MaximumReceiverRFOffset
        www.onsemi.com 28 Table28.CONTROLREGISTERMAP ReceiverParameterSet1 AX5043 Add Name Dir Ret Reset Bit
        Description 7 6 5 4 3 2 1 0 */
#define AX_MAXRFOFFSET1 0x010a /*RW R 00010110 MAXRFOFFSET(15:8) MaximumReceiverRFOffset */
#define AX_MAXRFOFFSET0 0x010b /*RW R 10000111 MAXRFOFFSET(7:0) MaximumReceiverRFOffset */
#define AX_FSKDMAX1 0x010c /*RW R 00000000 FSKDEVMAX(15:8) FourFSKRxDeviation */
#define AX_FSKDMAX0 0x010d /*RW R 10000000 FSKDEVMAX(7:0) FourFSKRxDeviation */
#define AX_FSKDMIN1 0x010e /*RW R 11111111 FSKDEVMIN(15:8) FourFSKRxDeviation */
#define AX_FSKDMIN0 0x010f /*RW R 10000000 FSKDEVMIN(7:0) FourFSKRxDeviation */
#define AX_AFSKSPACE1 0x0110 /*RW R ----0000 - - - - AFSKSPACE(11:8) AFSKSpace(0)Frequency */
#define AX_AFSKSPACE0 0x0111 /*RW R 01000000 AFSKSPACE(7:0) AFSKSpace(0)Frequency */
#define AX_AFSKMARK1 0x0112 /*RW R ----0000 - - - - AFSKMARK(11:8) AFSKMark(1)Frequency */
#define AX_AFSKMARK0 0x0113 /*RW R 01110101 AFSKMARK(7:0) AFSKMark(1)Frequency */
#define AX_AFSKCTRL 0x0114 /*RW R ---00100 - - - AFSKSHIFT0(4:0) AFSKControl */
#define AX_AMPLFILTER 0x0115 /*RW R ----0000 - - - - AMPLFILTER(3:0) AmplitudeFilter */
#define AX_FREQUENCYLEAK 0x0116 /*RW R ----0000 - - - - FREQUENCYLEAK[3:0] BasebandFrequencyRecoveryLoopLeakiness */
#define AX_RXPARAMSETS 0x0117 /*RW R 00000000 RXPS3(1:0) RXPS2(1:0) RXPS1(1:0) RXPS0(1:0)
        ReceiverParameterSetIndirection */
#define AX_RXPARAMCURSET 0x0118 /*R R -------- - - - RXSI(2) RXSN(1:0) RXSI(1:0) ReceiverParameterCurrentSet
        ReceiverParameterSet0 */
#define AX_AGCGAIN0 0x0120 /*RW R 10110100 AGCDECAY0(3:0) AGCATTACK0(3:0) AGCSpeed */
#define AX_AGCTARGET0 0x0121 /*RW R 01110110 AGCTARGET0(7:0) AGCTarget */
#define AX_AGCAHYST0 0x0122 /*RW R -----000 - AGCAHYST0(2:0) AGCDigitalThresholdRange */
#define AX_AGCMINMAX0 0x0123 /*RW R -000-000 - AGCMAXDA0(2:0) - AGCMINDA0(2:0) AGCDigitalMin/MaxSetPoints */
#define AX_TIMEGAIN0 0x0124 /*RW R 11111000 TIMEGAIN0M TIMEGAIN0E TimingGain */
#define AX_DRGAIN0 0x0125 /*RW R 11110010 DRGAIN0M DRGAIN0E DataRateGain */
#define AX_PHASEGAIN0 0x0126 /*RW R 11--0011 FILTERIDX0(1:0) - - PHASEGAIN0(3:0) FilterIndex,PhaseGain */
#define AX_FREQGAINA0 0x0127 /*RW R 00001111 FREQLIM0 FREQMODULO0 FREQHALFMOD0 FREQAMPLGATE0 FREQGAINA0(3:0)
        FrequencyGainA */
#define AX_FREQGAINB0 0x0128 /*RW R 00-11111 FREQFREEZE0 FREQAVG0 - FREQGAINB0(4:0) FrequencyGainB */
#define AX_FREQGAINC0 0x0129 /*RW R ---01010 - - - FREQGAINC0(4:0) FrequencyGainC */
#define AX_FREQGAIND0 0x012a /*RW R 0--01010 RFFREQFREEZE0 - - FREQGAIND0(4:0) FrequencyGainD */
#define AX_AMPLGAIN0 0x012b /*RW R 01--0110 AMPLAVG AMPLAGC - - AMPLGAIN0(3:0) AmplitudeGain */
#define AX_FREQDEV10 0x012c /*RW R ----0000 - - - - FREQDEV0(11:8) ReceiverFrequencyDeviation */
#define AX_FREQDEV00 0x012d /*RW R 00100000 FREQDEV0(7:0) ReceiverFrequencyDeviation */
#define AX_FOURFSK0 0x012e /*RW R ---10110 - - - DEVUPDATE0 DEVDECAY0(3:0) FourFSKControl */
#define AX_BBOFFSRES0 0x012f /*RW R 10001000 RESINTB0(3:0) RESINTA0(3:0) BasebandOffsetCompensationResistors */
#define AX_AGCGAIN1 0x0130 /*RW R 10110100 AGCDECAY1(3:0) AGCATTACK1(3:0) AGCSpeed */
#define AX_AGCTARGET1 0x0131 /*RW R 01110110 AGCTARGET1(7:0) AGCTarget */
#define AX_AGCAHYST1 0x0132 /*RW R -----000 - AGCAHYST1(2:0) AGCDigitalThresholdRange */
#define AX_AGCMINMAX1 0x0133 /*RW R -000-000 - AGCMAXDA1(2:0) - AGCMINDA1(2:0) AGCDigitalMin/MaxSetPoints */
#define AX_TIMEGAIN1 0x0134 /*RW R 11110110 TIMEGAIN1M TIMEGAIN1E TimingGain */
#define AX_DRGAIN1 0x0135 /*RW R 11110001 DRGAIN1M DRGAIN1E DataRateGain */
#define AX_PHASEGAIN1 0x0136 /*RW R 11--0011 FILTERIDX1(1:0) - - PHASEGAIN1(3:0) FilterIndex,PhaseGain www.onsemi.com
        29 Table28.CONTROLREGISTERMAP ReceiverParameterSet3 AX5043 Add Name Dir Ret Reset Bit Description 7 6 5 4 3 2
        1 0 */
#define AX_FREQGAINA1 0x0137 /*RW R 00001111 FREQLIM1 FREQMODULO1 FREQHALFMOD1 FREQAMPLGATE1 FREQGAINA1(3:0)
        FrequencyGainA */
#define AX_FREQGAINB1 0x0138 /*RW R 00-11111 FREQFREEZE1 FREQAVG1 - FREQGAINB1(4:0) FrequencyGainB */
#define AX_FREQGAINC1 0x0139 /*RW R ---01011 - - - FREQGAINC1(4:0) FrequencyGainC */
#define AX_FREQGAIND1 0x013a /*RW R 0--01011 RFFREQFREEZE1 - - FREQGAIND1(4:0) FrequencyGainD */
#define AX_AMPLGAIN1 0x013b /*RW R 01--0110 AMPLAVG1 AMPL1AGC1 - - AMPLGAIN1(3:0) AmplitudeGain */
#define AX_FREQDEV11 0x013c /*RW R ----0000 - - - - FREQDEV1(11:8) ReceiverFrequencyDeviation */
#define AX_FREQDEV01 0x013d /*RW R 00100000 FREQDEV1(7:0) ReceiverFrequencyDeviation */
#define AX_FOURFSK1 0x013e /*RW R ---11000 - - - DEVUPDATE1 DEVDECAY1(3:0) FourFSKControl */
#define AX_BBOFFSRES1 0x013f /*RW R 10001000 RESINTB1(3:0) RESINTA1(3:0) BasebandOffsetCompensationResistors
        ReceiverParameterSet2 */
#define AX_AGCGAIN2 0x0140 /*RW R 11111111 AGCDECAY2(3:0) AGCATTACK2(3:0) AGCSpeed */
#define AX_AGCTARGET2 0x0141 /*RW R 01110110 AGCTARGET2(7:0) AGCTarget */
#define AX_AGCAHYST2 0x0142 /*RW R -----000 - AGCAHYST2(2:0) AGCDigitalThresholdRange */
#define AX_AGCMINMAX2 0x0143 /*RW R -000-000 - AGCMAXDA2(2:0) - AGCMINDA2(2:0) AGCDigitalMin/MaxSetPoints */
#define AX_TIMEGAIN2 0x0144 /*RW R 11110101 TIMEGAIN2M TIMEGAIN2E TimingGain */
#define AX_DRGAIN2 0x0145 /*RW R 11110000 DRGAIN2M DRGAIN2E DataRateGain */
#define AX_PHASEGAIN2 0x0146 /*RW R 11--0011 FILTERIDX2(1:0) - - PHASEGAIN2(3:0) FilterIndex,PhaseGain */
#define AX_FREQGAINA2 0x0147 /*RW R 00001111 FREQLIM2 FREQMODULO2 FREQHALFMOD2 FREQAMPLGATE2 FREQGAINA2(3:0)
        FrequencyGainA */
```

```
#define AX_FREQGAINB2 0x0148 /*RW R 00-11111 FREQFREEZE2 FREQAVG2 - FREQGAINB2(4:0) FrequencyGainB */
#define AX_FREQGAINC2 0x0149 /*RW R ---01101 - - - FREQGAINC2(4:0) FrequencyGainC */
#define AX_FREQGAIND2 0x014a /*RW R 0--01101 RFFREQFREEZE2 - - FREQGAIND2(4:0) FrequencyGainD */
#define AX_AMPLGAIN2 0x014b /*RW R 01--0110 AMPLAVG2 AMPLAGC2 - - AMPLGAIN2(3:0) AmplitudeGain */
#define AX_FREQDEV12 0x014c /*RW R ----0000 - - - - FREQDEV2(11:8) ReceiverFrequencyDeviation */
#define AX_FREQDEV02 0x014d /*RW R 00100000 FREQDEV2(7:0) ReceiverFrequencyDeviation */
#define AX_FOURFSK2 0x014e /*RW R ---11010 - - - DEVUPDATE2 DEVDECAY2(3:0) FourFSKControl */
#define AX_BBOFFSRES2 0x014f /*RW R 10001000 RESINTB2(3:0) RESINTA2(3:0) BasebandOffsetCompensationResistors */
#define AX_AGCGAIN3 0x0150 /*RW R 11111111 AGCDECAY3(3:0) AGCATTACK3(3:0) AGCSpeed */
#define AX_AGCTARGET3 0x0151 /*RW R 01110110 AGCTARGET3(7:0) AGCTarget */
#define AX_AGCAHYST3 0x0152 /*RW R -----000 - AGCAHYST3(2:0) AGCDigitalThresholdRange */
#define AX_AGCMINMAX3 0x0153 /*RW R -000-000 - AGCMAXDA3(2:0) - AGCMINDA3(2:0) AGCDigitalMin/MaxSetPoints */
#define AX_TIMEGAIN3 0x0154 /*RW R 11110101 TIMEGAIN3M TIMEGAIN3E TimingGain */
#define AX_DRGAIN3 0x0155 /*RW R 11110000 DRGAIN3M DRGAIN3E DataRateGain */
#define AX_PHASEGAIN3 0x0156 /*RW R 11--0011 FILTERIDX3(1:0) - - PHASEGAIN3(3:0) FilterIndex,PhaseGain */
#define AX_FREQGAINA3 0x0157 /*RW R 00001111 FREQLIM3 FREQMODULO3 FREQHALFMOD3 FREQAMPLGATE3 FREQGAINA3(3:0)
         FrequencyGainA */
#define AX_FREQGAINB3 0x0158 /*RW R 00-11111 FREQFREEZE3 FREQAVG3 - FREQGAINB3(4:0) FrequencyGainB */
#define AX_FREQGAINC3 0x0159 /*RW R ---01101 - - - FREQGAINC3(4:0) FrequencyGainC www.onsemi.com 30
         Table28.CONTROLREGISTERMAP TransmitterParameters AX5043 Add Name Dir Ret Reset Bit Description 7 6 5 4 3 2 1 0
         */
#define AX_FREQGAIND3 0x015a /*RW R 0--01101 RFFREQFREEZE3 - - FREQGAIND3(4:0) FrequencyGainD */
#define AX_AMPLGAIN3 0x015b /*RW R 01--0110 AMPLAVG3 AMPLAGC3 - - AMPLGAIN3(3:0) AmplitudeGain */
#define AX_FREQDEV13 0x015c /*RW R ----0000 - - - - FREQDEV3(11:8) ReceiverFrequencyDeviation */
#define AX_FREQDEV03 0x015d /*RW R 00100000 FREQDEV3(7:0) ReceiverFrequencyDeviation */
#define AX_FOURFSK3 0x015e /*RW R ---11010 - - - DEVUPDATE3 DEVDECAY3(3:0) FourFSKControl */
#define AX_BBOFFSRES3 0x015f /*RW R 10001000 RESINTB3(3:0) RESINTA3(3:0) BasebandOffsetCompensationResistors */
#define AX_MODCFGF 0x0160 /*RW R ------00 - - - - - - FREQSHAPE ModulatorConfigurationF */
#define AX_FSKDEV2 0x0161 /*RW R 00000000 FSKDEV(23:16) FSKFrequencyDeviation */
#define AX_FSKDEV1 0x0162 /*RW R 00001010 FSKDEV(15:8) FSKFrequencyDeviation */
#define AX_FSKDEV0 0x0163 /*RW R 00111101 FSKDEV(7:0) FSKFrequencyDeviation */
#define AX_MODCFGA 0x0164 /*RW R 0000-101 BROWNGATE PTTLCKGATE SLOWRAMP - AMPLSHAPE TXSE TXDIFF ModulatorConfigurationA
         */
#define AX_TXRATE2 0x0165 /*RW R 00000000 TXRATE(23:16) TransmitterBitrate */
#define AX_TXRATE1 0x0166 /*RW R 00101000 TXRATE(15:8) TransmitterBitrate */
#define AX_TXRATE0 0x0167 /*RW R 11110110 TXRATE(7:0) TransmitterBitrate */
#define AX_TXPWRCOEFFA1 0x0168 /*RW R 00000000 TXPWRCOEFFA(15:8) TransmitterPredistortionCoefficientA */
#define AX_TXPWRCOEFFA0 0x0169 /*RW R 00000000 TXPWRCOEFFA(7:0) TransmitterPredistortionCoefficientA */
#define AX_TXPWRCOEFFB1 0x016a /*RW R 00001111 TXPWRCOEFFB(15:8) TransmitterPredistortionCoefficientB */
#define AX_TXPWRCOEFFB0 0x016b /*RW R 11111111 TXPWRCOEFFB(7:0) TransmitterPredistortionCoefficientB */
#define AX_TXPWRCOEFFC1 0x016c /*RW R 00000000 TXPWRCOEFFC(15:8) TransmitterPredistortionCoefficientC */
#define AX_TXPWRCOEFFC0 0x016d /*RW R 00000000 TXPWRCOEFFC(7:0) TransmitterPredistortionCoefficientC */
#define AX_TXPWRCOEFFD1 0x016e /*RW R 00000000 TXPWRCOEFFD(15:8) TransmitterPredistortionCoefficientD */
#define AX_TXPWRCOEFFD0 0x016f /*RW R 00000000 TXPWRCOEFFD(7:0) TransmitterPredistortionCoefficientD */
#define AX_TXPWRCOEFFE1 0x0170 /*RW R 00000000 TXPWRCOEFFE(15:8) TransmitterPredistortionCoefficientE */
#define AX_TXPWRCOEFFE0 0x0171 /*RW R 00000000 TXPWRCOEFFE(7:0) TransmitterPredistortionCoefficientE PLLParameters
         CrystalOscillator */
#define AX_PLLVCOI 0x0180 /*RW R 0-010010 VCOIE - VCOI(5:0) VCOCurrent */
#define AX_PLLVCOIR 0x0181 /*RW R -------- - - VCOIR(5:0) VCOCurrentReadback */
#define AX_PLLLOCKDET 0x0182 /*RW R -----011 LOCKDETDLYR - - - LOCKDETDLYM LOCKDETDLY PLLLockDetectDelay */
#define AX_PLLRNGCLK 0x0183 /*RW R -----011 - - - - - - PLLRNGCLK(2:0) PLLRangingClock */
#define AX_XTALCAP 0x0184 /*RW R 00000000 XTALCAP(7:0) CrystalOscillatorLoadCapacitance www.onsemi.com 31
         Table28.CONTROLREGISTERMAP Baseband MACLayerParametersPacketFormat AX5043 Add Name Dir Ret Reset Bit
         Description 7 6 5 4 3 2 1 0 */
#define AX_BBTUNE 0x0188 /*RW R ---01001 - - - BBTUNERUN BBTUNE(3:0) BasebandTuning */
#define AX_BBOFFSCAP 0x0189 /*RW R -111-111 - CAPINTB(2:0) - CAPINTA(2:0) BasebandOffsetCompensationCapacitors */
#define AX_PKTADDRCFG 0x0200 /*RW R 001-0000 MSBFIRST CRCSKIPFIRST FECSYNCDIS - ADDRPOS(3:0) PacketAddressConfig */
#define AX_PKTLENCFG 0x0201 /*RW R 00000000 LENBITS(3:0 LENPOS(3:0) PacketLengthConfig */
#define AX_PKTLENOFFSET 0x0202 /*RW R 00000000 LENOFFSET(7:0) PacketLengthOffset */
#define AX_PKTMAXLEN 0x0203 /*RW R 00000000 MAXLEN(7:0) PacketMaximumLength */
#define AX_PKTADDR3 0x0204 /*RW R 00000000 ADDR(31:24) PacketAddress3 */
#define AX_PKTADDR2 0x0205 /*RW R 00000000 ADDR(23:16) PacketAddress2 */
#define AX_PKTADDR1 0x0206 /*RW R 00000000 ADDR(15:8) PacketAddress1 */
#define AX_PKTADDR0 0x0207 /*RW R 00000000 ADDR(7:0) PacketAddress0 */
#define AX_PKTADDRMASK3 0x0208 /*RW R 00000000 ADDRMASK(31:24) PacketAddressMask1 */
#define AX_PKTADDRMASK2 0x0209 /*RW R 00000000 ADDRMASK(23:16) PacketAddressMask0 */
#define AX_PKTADDRMASK1 0x020a /*RW R 00000000 ADDRMASK(15:8) PacketAddressMask1 */
#define AX_PKTADDRMASK0 0x020b /*RW R 00000000 ADDRMASK(7:0) PacketAddressMask0 PatternMatch */
#define AX_MATCH0PAT3 0x0210 /*RW R 00000000 MATCH0PAT(31:24) PatternMatchUnit0,Pattern */
#define AX_MATCH0PAT2 0x0211 /*RW R 00000000 MATCH0PAT(23:16) PatternMatchUnit0,Pattern */
#define AX_MATCH0PAT1 0x0212 /*RW R 00000000 MATCH0PAT(15:8) PatternMatchUnit0,Pattern */
#define AX_MATCH0PAT0 0x0213 /*RW R 00000000 MATCH0PAT(7:0) PatternMatchUnit0,Pattern */
#define AX_MATCH0LEN 0x0214 /*RW R 0--00000 MATCH0RAW - - MATCH0LEN PatternMatchUnit0,PatternLength */
#define AX_MATCH0MIN 0x0215 /*RW R ---00000 - - - MATCH0MIN PatternMatchUnit0,MinimumMatch */
#define AX_MATCH0MAX 0x0216 /*RW R ---11111 - - - MATCH0MAX PatternMatchUnit0,MaximumMatch */
#define AX_MATCH1PAT1 0x0218 /*RW R 00000000 MATCH1PAT(15:8) PatternMatchUnit1,Pattern */
#define AX_MATCH1PAT0 0x0219 /*RW R 00000000 MATCH1PAT(7:0) PatternMatchUnit1,Pattern */
#define AX_MATCH1LEN 0x021c /*RW R 0---0000 MATCH1RAW - - - MATCH1LEN PatternMatchUnit1,PatternLength */
#define AX_MATCH1MIN 0x021d /*RW R ----0000 - - - - MATCH1MIN PatternMatchUnit1,MinimumMatch */
#define AX_MATCH1MAX 0x021e /*RW R ----1111 - - - - MATCH1MAX PatternMatchUnit1,MaximumMatch PacketController */
#define AX_TMGTXBOOST 0x0220 /*RW R 00110010 TMGTXBOOSTE TMGTXBOOSTM TransmitPLLBoostTime */
#define AX_TMGTXSETTLE 0x0221 /*RW R 00001010 TMGTXSETTLEE TMGTXSETTLEM TransmitPLL(postBoost)SettlingTime */
#define AX_TMGRXBOOST 0x0223 /*RW R 00110010 TMGRXBOOSTE TMGRXBOOSTM ReceivePLLBoostTime */
#define AX_TMGRXSETTLE 0x0224 /*RW R 00010100 TMGRXSETTLEE TMGRXSETTLEM ReceivePLL(postBoost)SettlingTime */
#define AX_TMGRXOFFSACQ 0x0225 /*RW R 01110011 TMGRXOFFSACQE TMGRXOFFSACQM ReceiveBasebandDCOffsetAcquisitionTime */
#define AX_TMGRXCOARSEAGC 0x0226 /*RW R 00111001 TMGRXCOARSEAGCE TMGRXCOARSEAGCM ReceiveCoarseAGCTime www.onsemi.com 32
         Table28.CONTROLREGISTERMAP SpecialFunctionsGeneralPurposeADC LowPowerOscillatorCalibration AX5043 Add Name Dir
         Ret Reset Bit Description 7 6 5 4 3 2 1 0 */
```

```
#define AX_TMGRXAGC 0x0227 /*RW R 00000000 TMGRXAGCE TMGRXAGCM ReceiverAGCSettlingTime */
#define AX_TMGRXRSSI 0x0228 /*RW R 00000000 TMGRXRSSIE TMGRXRSSIM ReceiverRSSISettlingTime */
#define AX_TMGRXPREAMBLE1 0x0229 /*RW R 00000000 TMGRXPREAMBLE1E TMGRXPREAMBLE1M ReceiverPreamble1Timeout */
#define AX_TMGRXPREAMBLE2 0x022a /*RW R 00000000 TMGRXPREAMBLE2E TMGRXPREAMBLE2M ReceiverPreamble2Timeout */
#define AX_TMGRXPREAMBLE3 0x022b /*RW R 00000000 TMGRXPREAMBLE3E TMGRXPREAMBLE3M ReceiverPreamble3Timeout */
#define AX_RSSIREFERENCE 0x022c /*RW R 00000000 RSSIREFERENCE RSSIOffset */
#define AX_RSSIABSTHR 0x022d /*RW R 00000000 RSSIABSTHR RSSIAbsoluteThreshold */
#define AX_BGNDRSSIGAIN 0x022e /*RW R ----0000 - - - - BGNDRSSIGAIN BackgroundRSSIAveragingTimeConstant */
#define AX_BGNDRSSITHR 0x022f /*RW R --000000 - - BGNDRSSITHR BackgroundRSSIRelativeThreshold */
#define AX_PKTCHUNKSIZE 0x0230 /*RW R ----0000 - - - - PKTCHUNKSIZE(3:0) PacketChunkSize */
#define AX_PKTMISCFLAGS 0x0231 /*RW R ---00000 - - - WORMULTIPKT AGCSETTLDET BGNDRSSI RXAGCCLK RXRSSICLK
         PacketControllerMiscellaneousFlags */
#define AX_PKTSTOREFLAGS 0x0232 /*RW R -0000000 - STANTRSSI STCRCB STRSSI STDR STRFOFFS STFOFFS STTIMER
         PacketControllerStoreFlags */
#define AX_PKTACCEPTFLAGS 0x0233 /*RW R --000000 - - ACCPTLRGP ACCPTSZF ACCPTADDRF ACCPTCRCF ACCPTABRT ACCPTRESIDUE
         PacketControllerAcceptFlags */
#define AX_GPADCCTRL 0x0300 /*RW R --000000 BUSY - 0 0 0 GPADC13 CONT CHISOL GeneralPurposeADCControl */
#define AX_GPADCPERIOD 0x0301 /*RW R 00111111 GPADCPERIOD(7:0) GPADCSamplingPeriod */
#define AX_GPADC13VALUE1 0x0308 /*R -------- - - - - - - GPADC13VALUE(9:8) GPADC13Value */
#define AX_GPADC13VALUE0 0x0309 /*R -------- GPADC13VALUE(7:0) GPADC13Value */
#define AX_LPOSCCONFIG 0x0310 /*RW 00000000 LPOSCOSCINVERT LPOSCOSCDOUBLE LPOSCCALIBR LPOSCCALIBF LPOSCIRQR LPOSCIRQF
         LPOSCFAST LPOSCENA LowPowerOscillatorConfiguration 31 LPOSCSTATUS R -------- - - - - - - LPOSCIRQ LPOSCEDGE
         LowPowerOscillatorStatus */
#define AX_LPOSCKFILT1 0x0312 /*RW 00100000 LPOSCKFILT(15:8) LowPowerOscillatorCalibrationFilterConstant */
#define AX_LPOSCKFILT0 0x0313 /*RW 11000100 LPOSCKFILT(7:0) LowPowerOscillatorCalibrationFilterConstant */
#define AX_LPOSCREF1 0x0314 /*RW 01100001 LPOSCREF(15:8) LowPowerOscillatorCalibrationReference */
#define AX_LPOSCREF0 0x0315 /*RW 10101000 LPOSCREF(7:0) LowPowerOscillatorCalibrationReference */
#define AX_LPOSCFREQ1 0x0316 /*RW 00000000 LPOSCFREQ(9:2) LowPowerOscillatorCalibrationFrequency */
#define AX_LPOSCFREQ0 0x0317 /*RW 0000---- LPOSCFREQ(1:-2) - - - - LowPowerOscillatorCalibrationFrequency */
#define AX_LPOSCPER1 0x0318 /*RW -------- LPOSCPER(15:8) LowPowerOscillatorCalibrationPeriod */
#define AX_LPOSCPER0 0x0319 /*RW -------- LPOSCPER(7:0) LowPowerOscillatorCalibrationPeriod */
#define AX_DACVALUE1 0x0330 /*RW R ----0000 - - - - DACVALUE(11:8) DACValue */
#define AX_DACVALUE0 0x0331 /*RW R 00000000 DACVALUE(7:0) DACValue */
#define AX_DACCONFIG 0x0332 /*RW R 00--0000 DACPWM DACCLKX2 - - DACINPUT(3:0) DACConfiguration */

#define AX_PWRMODE_MODE_POWERDOWN 0x0
#define AX_PWRMODE_MODE_DEEPSLEEP 0x1
#define AX_PWRMODE_MODE_STANDBY   0x5
#define AX_PWRMODE_MODE_FIFOON    0x7
#define AX_PWRMODE_MODE_SYNTHRX   0x8
#define AX_PWRMODE_MODE_FULLRX    0x9
#define AX_PWRMODE_MODE_WORRX     0xb
#define AX_PWRMODE_MODE_SYNTHTX   0xc
#define AX_PWRMODE_MODE_FULLTX    0xd

#define AX_PWRMODE_WDS   0x10
#define AX_PWRMODE_REFEN 0x20
#define AX_PWRMODE_XOEN  0x40
#define AX_PWRMODE_RST   0x80
```

# FREERTOSCONFIG.H

```
/*
    FreeRTOS V9.0.0 - Copyright (C) 2016 Real Time Engineers Ltd.
    All rights reserved

    VISIT http://www.FreeRTOS.org TO ENSURE YOU ARE USING THE LATEST VERSION.

    This file is part of the FreeRTOS distribution.

    FreeRTOS is free software; you can redistribute it and/or modify it under
    the terms of the GNU General Public License (version 2) as published by the
    Free Software Foundation >>>> AND MODIFIED BY <<<< the FreeRTOS exception.

    ***************************************************************************
    >>!   NOTE: The modification to the GPL is included to allow you to     !<<
    >>!   distribute a combined work that includes FreeRTOS without being   !<<
    >>!   obliged to provide the source code for proprietary components     !<<
    >>!   outside of the FreeRTOS kernel.                                   !<<
    ***************************************************************************

    FreeRTOS is distributed in the hope that it will be useful, but WITHOUT ANY
    WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS
    FOR A PARTICULAR PURPOSE.  Full license text is available on the following
    link: http://www.freertos.org/a00114.html

    ***************************************************************************
     *                                                                       *
     *    FreeRTOS provides completely free yet professionally developed,    *
     *    robust, strictly quality controlled, supported, and cross          *
     *    platform software that is more than just the market leader, it     *
     *    is the industry's de facto standard.                               *
     *                                                                       *
     *    Help yourself get started quickly while simultaneously helping     *
     *    to support the FreeRTOS project by purchasing a FreeRTOS           *
     *    tutorial book, reference manual, or both:                          *
     *    http://www.FreeRTOS.org/Documentation                              *
     *                                                                       *
    ***************************************************************************

    http://www.FreeRTOS.org/FAQHelp.html - Having a problem?  Start by reading
    the FAQ page "My application does not run, what could be wrong?".  Have you
    defined configASSERT()?

    http://www.FreeRTOS.org/support - In return for receiving this top quality
    embedded software for free we request you assist our global community by
    participating in the support forum.

    http://www.FreeRTOS.org/training - Investing in training allows your team to
    be as productive as possible as early as possible.  Now you can receive
    FreeRTOS training directly from Richard Barry, CEO of Real Time Engineers
    Ltd, and the world's leading authority on the world's leading RTOS.

    http://www.FreeRTOS.org/plus - A selection of FreeRTOS ecosystem products,
    including FreeRTOS+Trace - an indispensable productivity tool, a DOS
    compatible FAT file system, and our tiny thread aware UDP/IP stack.

    http://www.FreeRTOS.org/labs - Where new FreeRTOS products go to incubate.
    Come and try FreeRTOS+TCP, our new open source TCP/IP stack for FreeRTOS.

    http://www.OpenRTOS.com - Real Time Engineers ltd. license FreeRTOS to High
    Integrity Systems ltd. to sell under the OpenRTOS brand.  Low cost OpenRTOS
    licenses offer ticketed support, indemnification and commercial middleware.

    http://www.SafeRTOS.com - High Integrity Systems also provide a safety
    engineered and independently SIL3 certified version for use in safety and
    mission critical applications that require provable dependability.

    1 tab == 4 spaces!
*/

/*
 * The following #error directive is to remind users that a batch file must be
 * executed prior to this project being built.  Once it has been executed
 * remove the #error line below.
 */


#ifndef FREERTOS_CONFIG_H
#define FREERTOS_CONFIG_H

/*-----------------------------------------------------------
 * Application specific definitions.
 *
 * These definitions should be adjusted for your particular hardware and
 * application requirements.
```

```
 *
 * THESE PARAMETERS ARE DESCRIBED WITHIN THE 'CONFIGURATION' SECTION OF THE
 * FreeRTOS API DOCUMENTATION AVAILABLE ON THE FreeRTOS.org WEB SITE.
 *
 * See http://www.freertos.org/a00110.html.
 *----------------------------------------------------------*/

#define configUSE_PREEMPTION                                    1
#define configUSE_PORT_OPTIMISED_TASK_SELECTION         1
#define configUSE_IDLE_HOOK                                     0
#define configUSE_TICK_HOOK                                     0
#define configUSE_TRACE_FACILITY                        0
#define configUSE_16_BIT_TICKS                                  0
#define configCPU_CLOCK_HZ                                      ( ( unsigned long ) 50000000 ) /* Timer clock.
        */
#define configTICK_RATE_HZ                                      ( ( TickType_t ) 1000 )
#define configMAX_PRIORITIES                                    ( 8 )
#define configMINIMAL_STACK_SIZE                        ( ( unsigned short ) 128 )
#define configTOTAL_HEAP_SIZE                                   ( ( size_t ) 32768 )
#define configMAX_TASK_NAME_LEN                                 ( 16 )
#define configIDLE_SHOULD_YIELD                                 1
#define configGENERATE_RUN_TIME_STATS                   0
#define configUSE_MALLOC_FAILED_HOOK                    0

#define configCHECK_FOR_STACK_OVERFLOW                  0

/* Co-routine definitions. */
#define configUSE_CO_ROUTINES                                   0
#define configMAX_CO_ROUTINE_PRIORITIES                 ( 2 )

/* Mutexes */
#define configUSE_MUTEXES                                       1
#define configUSE_RECURSIVE_MUTEXES                     1

/* Semaphores */
#define configUSE_COUNTING_SEMAPHORES                   1

/* Timers */
#define configUSE_TIMERS                                        1
#define configTIMER_TASK_PRIORITY                       ( 2 )
#define configTIMER_QUEUE_LENGTH                        10
#define configTIMER_TASK_STACK_DEPTH                    ( 128 )

/* Set the following definitions to 1 to include the API function, or zero to exclude the API function. */
#define INCLUDE_vTaskPrioritySet                        1
#define INCLUDE_uxTaskPriorityGet                       1
#define INCLUDE_vTaskDelete                                     1
#define INCLUDE_vTaskCleanUpResources                   0
#define INCLUDE_vTaskSuspend                                    1
#define INCLUDE_xTaskResumeFromISR                      1
#define INCLUDE_vTaskDelayUntil                                 1
#define INCLUDE_vTaskDelay                                      1
#define INCLUDE_xTaskGetSchedulerState                  1
#define INCLUDE_uxTaskGetStackHighWaterMark   1

#define configASSERT( x ) if( ( x ) == pdFALSE ) { taskDISABLE_INTERRUPTS(); for( ;; ); }

#endif /* FREERTOS_CONFIG_H */
```

## ENDNOTES

[i] Internal Housekeeping Unit, the onboard processor typically responsible for overall control of a satellite's functions.
[ii] Advanced Satellite Communications and Exploration of Technology
[iii] http://www.amsat.org/mailman/listinfo/ascent
[iv] http://www.onsemi.com/PowerSolutions/product.do?id=AX5043
[v] Gaussian frequency shift keying, Minimum-shift keying, Frequency shift keying, Amplitude shift keying, Phase shift keying
[vi] ON Semiconductor, "AX5043 Advanced High Performance ASK and FSK Narrow-band Transceiver for 27 - 1050 MHz Range," AX5043 datasheet, Oct. 2015, p. 3
[vii] ON Semiconductor, "AX5043 Advanced High Performance ASK and FSK Narrow-band Transceiver for 27 - 1050 MHz Range," AX5043 datasheet, Oct. 2015, p. 34
[viii] ON Semiconductor, "AX5043 Advanced High Performance ASK and FSK Narrow-band Transceiver for 27 - 1050 MHz Range," AX5043 datasheet, Oct. 2015, p. 34
[ix] NXP Semiconductors, "Heterojunction Bipolar Transistor Technology (InGaP HBT) High Efficiency/Linearity Amplifier," MMZ09312BT1 datasheet, 2014, p.1
[x] NXP Semiconductors, "Heterojunction Bipolar Transistor Technology (InGaP HBT) High Efficiency/Linearity Amplifier," MMZ09312BT1 datasheet, 2014, p.2
[xi] Skyworks Solutions, "SKY13446-374LF: 0.1-6.0 GHz GaAs SPDT Switch", SKY13446-374LF datasheet, March 13, 2013, p. 1
[xii] Controller Area Network
[xiii] Maxim Integrated, "MAX 3051 +3.3V, 1Mbps, Low-Supply-Current CAN Transceiver," MAX3051 datasheet, Rev 3; 2/15, p. 1
[xiv] Maxim Integrated, "MAX 3051 +3.3V, 1Mbps, Low-Supply-Current CAN Transceiver," MAX3051 datasheet, Rev 3; 2/15, p. 8
[xv] Maxim Integrated, "MAX31725/MAX31726 +/- 0.5°C Local Temperature Sensors", MAX31725 datasheet, Rev 2; 12/14, p. 1
[xvi] Maxim Integrated, "MAX31725/MAX31726 +/- 0.5°C Local Temperature Sensors", MAX31725 datasheet, Rev 2; 12/14, p. 7
[xvii] http://www.ti.com/tool/LAUNCHXL2-570LC43
[xviii] http://www.ti.com/lsds/ti/tools-software/launchpads/boosterpacks/build/build.page
[xix] Gaussian Minimum Shift Keying
[xx] Texas Instruments, "TMS570LC4357 Hercules™ Microcontroller Based on the ARM® Cortex®-R Core", TMS570 Hercules™ datasheet, June 2016, p. 5
[xxi] http://www.ti.com/tool/CCSTUDIO
[xxii] http://www.onsemi.com/pub/Collateral/AND9356-D.PDF
[xxiii] http://www.onsemi.com/PowerSolutions/supportDoc.do?type=software
[xxiv] http://www.freertos.org/
[xxv] http://kicad-pcb.org/

# Evolution of the VITA 74 Standard (VNX) for "SpaceVNX" CubeSAT Applications


- Bill Ripley, KY5Q - (Embedded Systems Consultant, Trident Infosol / Brame Technologies), Chairman ANSI/VITA 74 Technical Working Group Bill.Ripley@Brame-Tech.com

- Patrick Collier - (Open Systems Thought Leader, Harris Corp / NAVAIR) Charles.p.collier@navy.mil



**ABSTRACT**

A new standard is working its way through the Standards Community. This standard defines the module functionality, interface and packaging concepts for Small Form Factor (SFF) embedded computers, I/O modules, power supplies, backplanes and chassis building blocks. This new standard was recently ratified by the VITA Standards Organization (VSO) and should be ratified by ANSI and released as a joint ANSI/VITA standard by the end of 2017. This new standard is referred to as "VITA 74", also known by its trade name as "VNX". The VITA 74.0 Compliant System Small Form Factor Module Base Standard has been created to meet the growing needs for improved Size, Weight and Power (SWaP) with a rugged, low cost, fast serial fabric interconnect based Plug-In Module, while leveraging many proven features of existing ANSI, VITA and other military and commercial standards.

A specific implementation of VITA 74 is being developed to address the specific needs of the CubeSAT and SmallSAT marketplace, and this implementation will be specifically referred to as "VITA 74.4" or "SpaceVNX". This paper will describe the essential elements of VITA 74, and discuss current plans to evolve the baseline VITA 74.0 standard to VITA 74.4, SpaceVNX.


**INTRODUCTION**

The VITA 74.0 standard describes both mechanical and electrical standards required to implement a Small Form Factor (SFF) system utilizing compliant VITA 74 modules. The standard addresses a need for a unified approach to small-scale systems to be used in a variety of MIL-Rugged applications. The standard encourages multiple vendors to supply components to be used in SFF systems at various levels; i.e. modules, backplanes, enclosures and complete solutions. The

goal is to allow vendor implementation flexibility, at the same time maximizing component interoperability.

To minimize development and schedule risk associated with creating a new standard, there was consensus from the developers to take advantage of previous work. The VITA 46 (VPX) and VITA 65 (OpenVPX) standards have been used as a basis for the VITA 74.0 base standard, and VITA 78 (SpaceVPX and SpaceVPX Lite) will be used as the basis for VITA 74.4 (SpaceVNX). This standard takes many of the conventions and leverages the signal definitions from its larger and higher complexity cousins into a Small Form Factor environment. The standard adopts the signal definitions from the larger standards, introduces a new high-speed low-cost connector, and reduces the number of serial fabric options. The result is a standard that has two primary serial fabrics, Ethernet and PCI-Express, but can evolve in the future to allow other high-speed fabric interconnects, such as Serial RapidIO and other similar fabrics which may be further developed. In the interest of keeping the system as simple to implement as possible, a single CPU root node is specified. There are no reasons why multiple CPU implementations cannot be accommodated and defined in a future “dot” standard.

The base standard gives details on the mechanical implementation of the system Plug-In Modules. Two module types are defined, 19mm and 12.5mm. Each type addresses a different set of functions. Fabrication of the module container is left to the vendor. In addition, details are provided for the module base carrier board (traditionally referred to as a backplane), including connector location, assembly hole locations and component keep-out areas. The 19mm module has ample room for a two-board stack with requisite heat spreaders. The inter-board connector selection is outside the scope of this standard. Suitable connectors are available from various manufactures to meet the stacking height requirement, with consideration given to board thickness and component heights.
Finally, even though the standard doesn’t specify a specific backplane implementation, details are provided as part of a reference design for a 4-slot backplane, available to VITA subscribers. This reference design may be used as a design starting point for a derivative implementation, with the specific implementation details defined by the application. Also, an example chassis design, with dimensional details, is available as a starting point from VITA. The chassis example may be reproduced in whole, or modified as required depending on the requirements.

## INTRODUCTION

The tactical embedded marketplace operates with unique requirements that drive SWaP (Size, Weight, and Power) to the extreme. Like the commercial counterpart, there are still requirements for high performance but at a much-reduced SWaP proposition. VITA 74 standard based products serve markets needing ruggedized products requiring Data Plane interconnect technologies that closely follow the industry state of the art. Switched serial technologies are available that provide significant benefits and scale over the long term. The benefits of these technologies are widely known with many products currently in the market incorporating one or more of these.

The VITA 74 standard provides a standardized electro-mechanical format for standardization of switched serial interconnects for ruggedized SFF applications. There are many candidate technologies for switched serial interconnects. These include Gigabit Ethernet and PCI Express. There is also the need for legacy I/O interfaces that provide connection to standard serial, discrete and analog devices.

The usage model for the Plug-In Modules described in this standard is very different than other Plug-In Modules in the VITA family of standards. Plug-In Modules are installed into an open frame backplane and then the chassis walls are attached, as opposed to a typical VPX or VME implementation where Plug-In Modules are installed into a fixed wall chassis.

## VITA 74 OBJECTIVES

The objectives of the VITA 74 standard are to:

- Define a 12.5mm and 19mm family of Small Form Factor Plug-In Modules
- Define a suitable high-speed connector family for use in Plug-In Modules
- Make provision for switched serial interconnects between Plug-In Modules
- Accommodate Open Standard technology for switched serial interconnects
- Make provision for additional I/O capability at the Plug-In Module level
- Define a mechanical standard for the Plug-In Modules

## Overview of the VITA 74 Family of Standards

The VITA 74 family of standards consists of this base standard, which defines physical features of compliant components; protocol layer standards, which define specific serial or parallel interconnects used in a system implementation;

and complimentary physical and/or electrical standards.

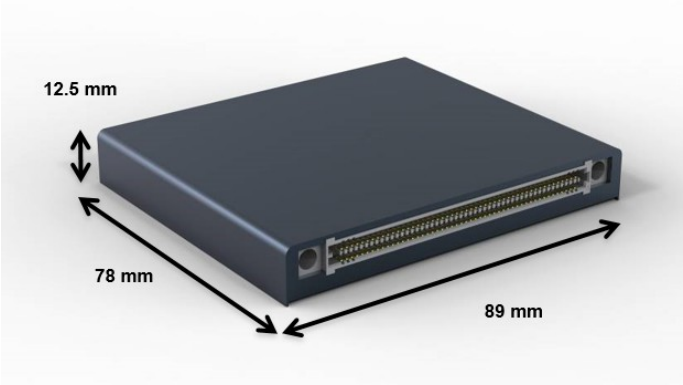
At the time of publication, the following dot standards have been allocated, but not yet published.

- VITA 74.0 is the base standard and the primary definition of the vision. The standard describes a Plug-In Module designed for backplane systems with one or more modules communicating over the backplane. This is the base standard that will be first in the series published by ANSI. For marketing purposes, VITA 74.0 has been branded as “VNX”.
- VITA 74.1: (Reserved)
- VITA 74.2 is the 74.0 rear transition module standard.
- VITA 74.3 is set of Slot Profiles for CPU, I/O, Storage and other specific modules. This standard will define pin assignments and signal characteristics that when used in conjunction with VITA 74.0, will serve to enhance module interoperability between vendors.
- VITA 74.4, also known as SpaceVNX, will be to generate additions to the baseline standards that will be equally applicable for high reliability and high availability Avionic and Flight Control applications, as is the case with Space applications. This standard is modeled after “SpaceVPX-Lite”, currently in work by VITA.
- VITA 74.5 is a standard for VNX with an Optical Connection. This standard will bring optics through the backplane using existing VNX connector with adjacent auxiliary MT Ferrule.
- VITA 74.6 is a standard for VNX with one or more RF Connections. This standard will bring RF through the backplane using existing VNX connector with adjacent RF connectors, as well as to provide standards for bringing RF from the front of the module.
- VITA 74.7 is a standard for VNX to facilitate High Performance Cooling and Packaging. This standard will utilize chassis-mounted wedge locks with VNX baseline modules for higher power applications when needed.

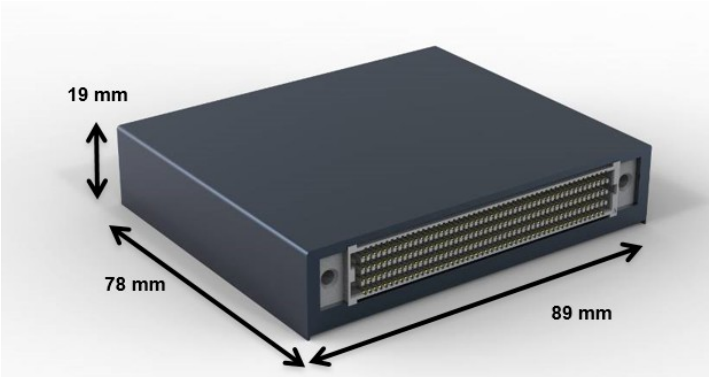
The VITA 74.0 base standard includes features drawn from IEEE 1101.2-1992, Conduction Cooled Plug-In Modules. These features are evident in the mechanical drawings provided.

The VITA 74.0 base standard defines physical and electrical features on the VITA 74 Compliant System Plug-In Module that enable high-speed communication in a compliant system. These features include a definition of two module sizes; a 78mm x 89mm x 19mm module and a 78mm x 89mm x 12.5mm module, both with high-speed differential and single ended signals. The electrical signal definitions, differential, single-ended, power and maintenance are defined in the base standard as interconnects. The base standard includes a standard fabric type, PCI-Express. The PCI-Express fabric acts as the main conduit for I/O internal to a system.

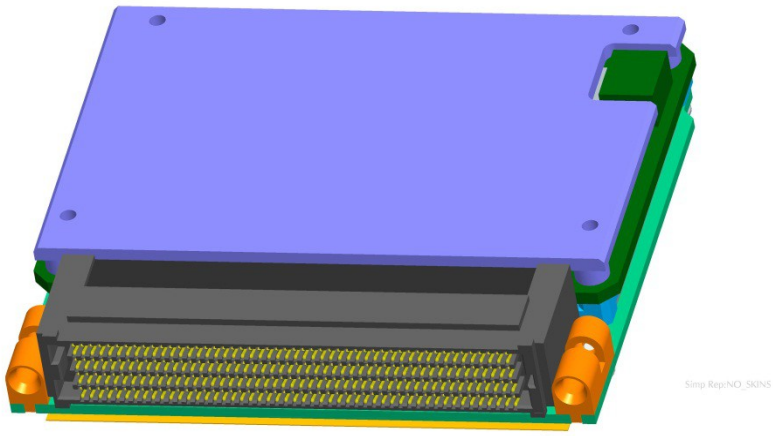
The base standard also defines alignment and keying features used to protect the connector system. The Plug-In Modules can only be oriented in one direction when being inserted into a slot. This keying is implemented in the module mechanical shell, its associated slot guide, the connector and its counterpart on the backplane.



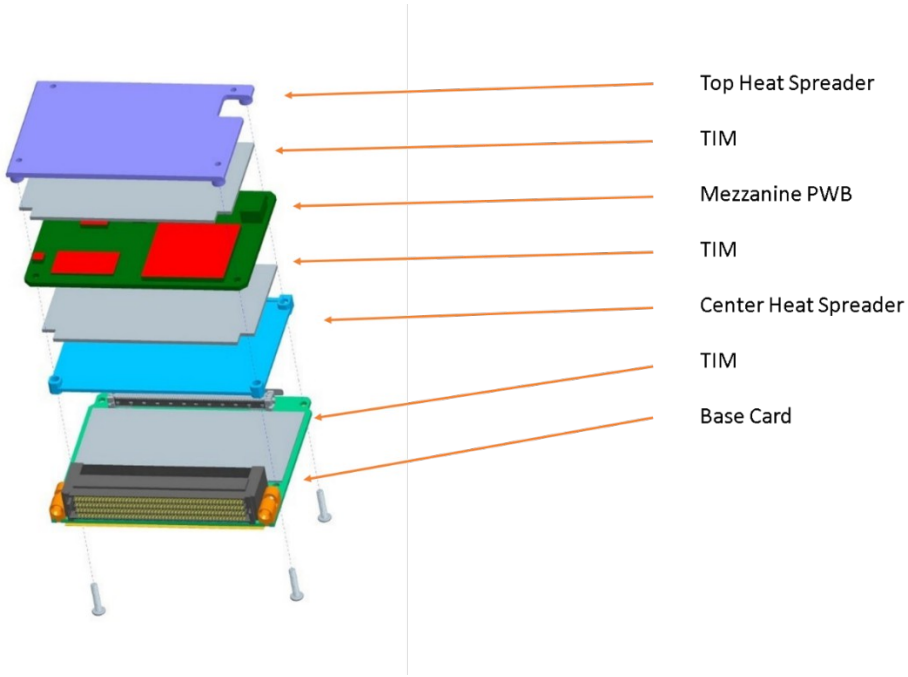

**12.5mm VITA 74 Compliant System Module**




**19mm VITA 74 Compliant System Module**



**VITA 74 Module without Cover**



## What makes a VITA 74 Module Compliant?

All VITA 74 compliant Plug-In Modules, backplanes, and systems incorporate the following major features:

- Connectors/connector sets that meet the requirements of this standard
- Keying and alignment mechanism for all slots that meet the requirements of this standard
- Support for the VITA 74-defined power rails VS1, VS2, VS3, VS4
- Support for the VITA 74-defined utility signals: Geographical Address pins, PCI Express Reference Clocks, System Management signals, Non-Volatile Memory Read Only (NVMRO) signal, SYSRESET*, Reserved Bussed signals, and Reserved for Future Use (RFU) single ended signals. The VITA 74.0 Base Standard reserves RFU signals; not the subsequent dot standards.
- Fabric provisioning for one or more of the various protocol layer standards, including the Base protocol, PCI Express

## VITA 74CubeSAT System Implementation Based on Reference Design

The purpose of this section is to describe how you can use modules that were designed in accordance with the VITA 74.0 base standard and integrate them into a typical conduction cooled system that would be appropriate for a CubeSAT, as well as an avionic, vetronic, or rugged industrial application.

*It should be noted that the images in this paper describe hardware that was intended for a rugged airborne or terrestrial application suitable for VITA 74.0. This paper is intended as a "thought piece" to discuss ways that this technology can be adapted in the VITA 74.4 SpaceVNX standard which is under development at the present time. To date, the VITA 74.4 SpaceVNX standard is still in development, but the time is right to make inputs to the designers of the standard to address the needs of the CubeSAT community.*

*It is recognized that certain changes will need to be made in the standard to optimize it for the CubeSAT marketplace, such as taking mass out of some of the metalwork, adding additional hold-down hardware to secure the modules into the chassis frame, and optimizing the thermal heat spreaders on the system.*

## VITA 74 System Reference Design

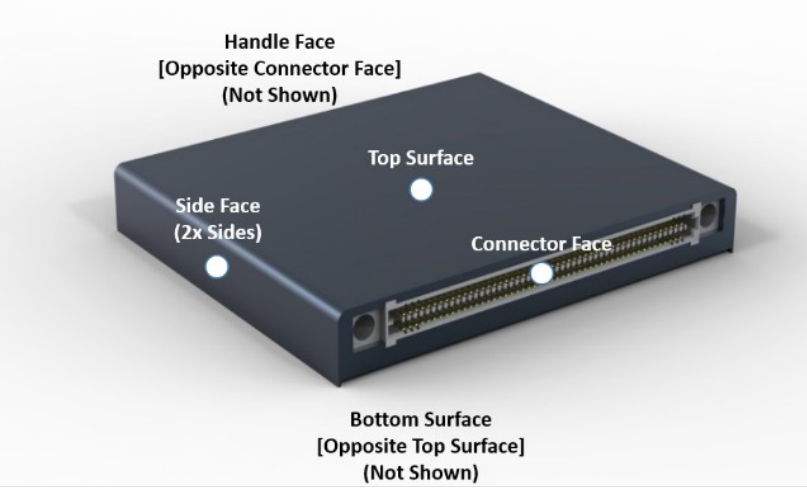
Along with the VITA 74.0 base standard, the authors of the standard have provided a reference design which is available to VITA members through the VSO website as well as those who purchase the standard. This reference design is intended to be used to illustrate how a typical system design using VITA 74.0 modules could be accomplished. This design shows many of the important constructs that are part of the standard, as well as many ideas on how the standard can be utilized to design and build complete system assemblies.

***It should be noted that the VITA 74 Reference Design has been designed to be ideally sized for a 1U CubeSAT application***.

See the VITA 74 reference design for further illustration and information.

## Definition of Terms

A VITA 74 Module has 6 primary faces which are defined as illustrated below. The same face nomenclature will be used for both the 12.5mm and 19mm modules defined in the standard, as well as discussed below.




VITA 74 Naming Convention of Module Faces

## VITA 74 Modules as Described in the VITA 74.0 Base Standard

Currently, two primary modules are described in the VITA 74.0 base standard. The module set consists of the 19mm module and the 12.5mm module.

Both the 12.5mm and the 19mm modules use the Samtec SeaRay series connector, with 200 contacts and 400 contacts respectively. The connectors have been qualified and shown to withstand the rigors of high vibration, extreme temperature environments. Both connectors have excellent frequency response characteristics, suitable for high speed signalling, up to and including RF frequencies.

Typically, a 19mm module will have two substantial boards (a base card and a mezzanine card) inside of its envelope. For many applications, such as an airborne mission computer, a 19mm module is generally some sort of processing element, i.e. a “high performance” “Single Board Computer” (SBC) or some other type of processor module, such as a Field Programmable Gate Array (FPGA) or Video Graphics Processing Unit (VGPU). A 19mm module can also be an I/O card that requires a lot of user I/O pins, or a board with oversize components such as 496 transformers, etc. For our CubeSAT example, we will assume that a 19mm module could be a crossband repeater with audio processing and control functions. A second 19mm module could house a Telemetry and Command & Control Radio.

The 12.5mm module is typically some sort of module for a low powered processor (such as a PIC or an ARM processor), data bus I/O, serial communications, analog I/O, GPS / Inertial Measurement Unit, or bulk persistent storage. Generally, this module has a substantial base card, and is sized in such a way that it may contain two or more “Micro Mezzanine” boards such as a MiniPCIe, mSATA, or other custom form factors.

## Hardware Configuration of Application Example

For purposes of this illustration, assume that the system described in this section will have the following card slot assignments and payload modules:

**VITA 74 CubeSAT Application Example Slot Assignments**

| Slot | Module Function | Module Size |
|---|---|---|
| 1 | Crossband Repeater & Controller | 19 mm |
| 2 | Telemetry & Command Processor | 19 mm |
| 3 | System Processor | 12.5 mm |
| 4 | Inertial Measurement & Power Control | 12.5 mm |

In addition to the four payload modules above, the system would also contain a few application specific modules whose characteristics are not specifically specified in this standard, but are left to the system designer to develop and implement. Because of the similarity of VITA 74 (VNX) to VITA 46 (VPX), there are a lot of designs used in VITA 46 / VITA 65 (OpenVPX) that are useable as references for a VITA 74 system implementation.

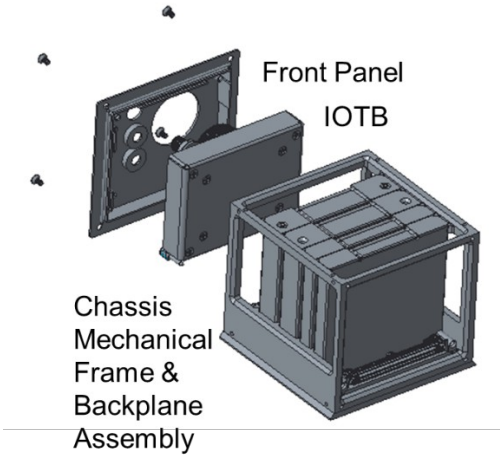
In a typical system, these modules and assemblies are defined and described below.

**Power Supply Unit**

The Power Supply Unit (PSU) interfaces with the primary power sources for the system (such as solar panels, batteries and thermo-electric generators) with the rest of the system. The PSU converts the supplied voltages to those specific voltages required by the application, as well as the power rails required by the standard.

**I/O Transition Board**

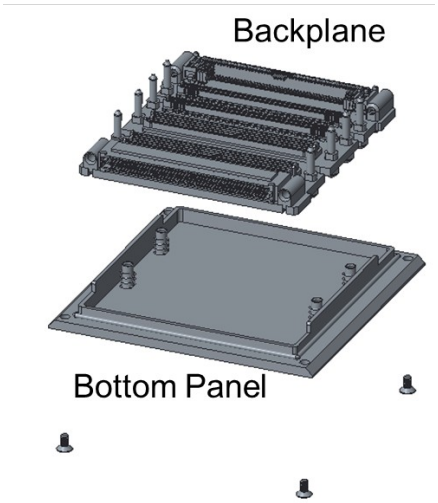
The I/O Transition Board (IOTB) is a printed wiring board (PWB) that functions as a connector panel for the interface to the outside world for those interfaces required for testing and I/O. As necessary, the IOTB also provides signal filtering and pin hardening. The IOTB plugs into the backplane through a connector with sufficient pins and bandwidth characteristics to allow the required signals to be routed from the backplane to the outside.




Application Example (Terrestrial System) IOTB with Front Panel & Chassis

**Backplane**

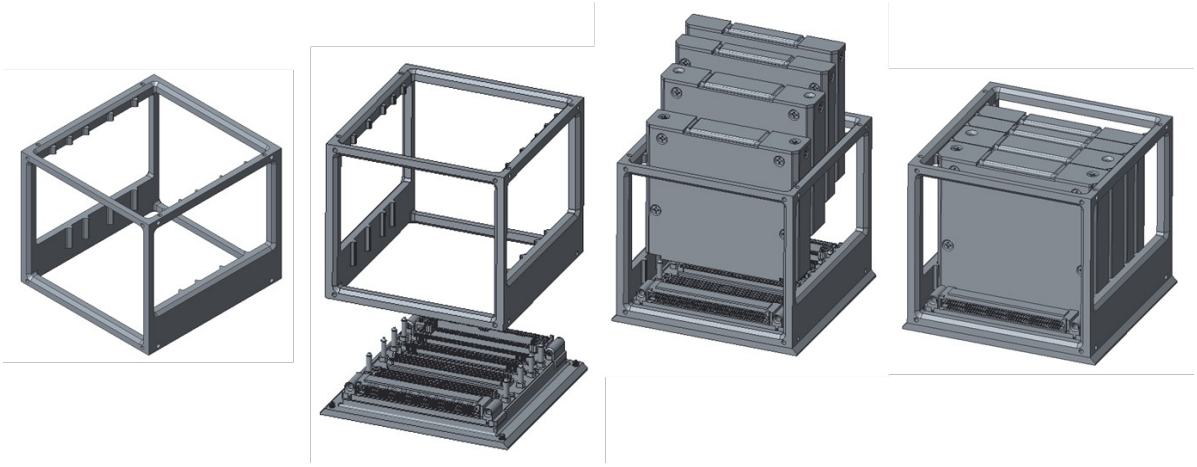
The Backplane is a printed wiring board which mounts into an associated Chassis Mechanical Frame Assembly (CMFA). The backplane allows electrical connections from each module slot to interconnect to other slots, as well as to other modules described in this section. As in a VPX system, the backplane is divided into 3 main sections including a Utility Plane which provides power and control signals to each module, a Serial Fabric Plane which routes data buses between slots, and User I/O Plane between the modules as required, as well as from the modules to the outside world via the IOTB. The backplane also has the mechanical guide pins which ensure that the module Connector Face and the backplane receptacle are properly aligned as they snap into position to mate the module and the backplane connectors.




VITA 74 Application Example Backplane & Bottom Panel

## Chassis Mechanical Frame Assembly (CMFA)

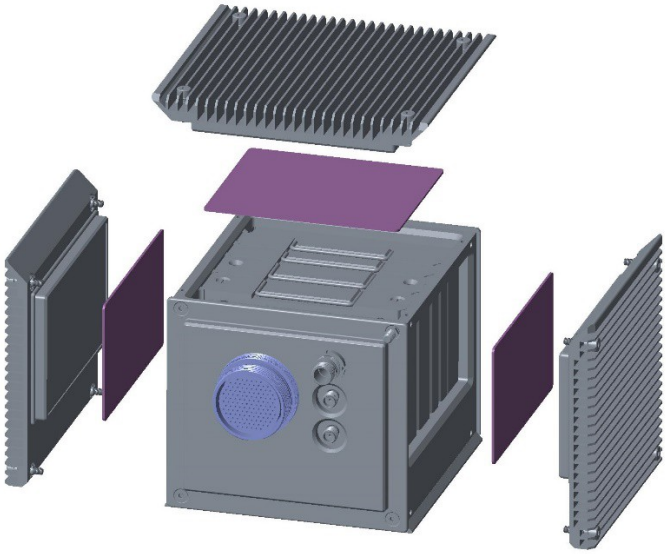
The CMFA is a mechanical structure which provides for correct card alignment and spacing in accordance with the VITA 74.0 standard. The CMFA is secured to the backplane. During assembly and operation, the CMFA holds all the modules in the correct orientation as the system is assembled and the side panels (which in a CubeSAT application would mount to the solar panels) are installed.



VITA 74 Application Example Chassis Mechanical Frame, Backplane & Payload Cards

## Side Panels

The Side Panels are the metal plates which mount to the sides and top of a typical VITA 74 system. The size of the fins and the capacity of the plates to conduct heat and spread it around the CubeSAT will drive the total power dissipation capability of the entire system. The side panels are in direct contact with the Side Faces of the VITA 74.0 module, through a properly sized piece of Thermal Interface Material (TIM).



VITA 74 Reference Chassis Design Exploded View

## Assembly of a VITA 74 System

VITA 74 is a Small Form Factor system which is designed to be light weight, optimally sized, and generally low power. To accomplish these ends, several design decisions were made early in the definition process that ultimately help facilitate the Space, Weight, and Power (SWaP) goals of the system designers.

One of the key attributes of the module design driven by the VITA 74.0 base standard is that the modules do not require locking devices (generically known as Wedge Locks) to hold the modules in place or to facilitate thermal heat transfer. Instead, the modules are inserted into the CMFA / Backplane Assembly, in a manner that the module's Connector Face mates with the Backplane. When mated correctly, an audible "Snap" will be heard as the Plugs and Receptacles correctly join. The Backplane connectors and guide pins, as well as the keying

notches in the CMFA keep the cards in the correct orientation and provide for optimal spacing. After all cards are inserted, Thermal Interface Material (TIM) is cut into thin strips and put on the two Side Faces and the Handle Face of each VITA 74 module. Please note here that the Top Surface and the Bottom Surface do not contribute to taking heat off the module in a typical back-to-back slotted chassis design. No TIM is to be applied to either of these surfaces for a "conventional" module installation. The system has a small gap between modules when the modules are positioned in the system correctly.

After the TIM is positioned on all the three designated faces on the module, the chassis side cooling panels and the top cooling panel are installed. General practice has been to install the first side cooling panel using the provided screws, snugging the screws only finger tight. The opposite side cooling panel is installed the same way, still tightening the screws only finger tight. Last, the remaining side (the top cooling panel) is installed, with the screws finger tight. After all three sides are loosely installed, then the screws on the cooling panels are all sequentially tightened in such a way to cause the entire chassis cooling fin assemblies to compress evenly around the CMFA, forming a dense block of modules, finned heat spreaders, and TIM. It has been shown that the thermal transmissibility from any module, through properly cut and applied TIM, to the side cooling panels with fins is within 5% to 10% of an equivalent system modeled with wedge locks.

## Conclusion

While many larger satellites have been built using standards based hardware (such as VME, VPX and more recently, SpaceVPX), embedded computing standards have not really been applied to the SmallSAT and CubeSAT space, with the exception of adaptations of PC/104, a standard generally used for convection cooled industrial applications. VITA 74 provides an opportunity to bring Small Form Factor, conduction cooled, highly reliable MIL-Rugged hardware into the mix of technologies to be considerred for future CubeSAT designs.

# Fox-1E (RadFxSat-2) Telemetry

*Burns Fisher, W2BFJ*

*w2bfj@amsat.org*

## Abstract

AMSAT satellites Fox-1A through Fox-1D (the FM Foxes) all do or will send telemetry at about 200 bits-per-second using a technique called Data Under Voice (DUV). DUV uses Frequency Shift Keying (FSK) to carry data in the baseband spectrum below about 300Hz. Fox-1E (officially known as RadFxSat-2), while using many of the same hardware components, contains a linear transponder with 1200 bits-per-second telemetry sent as Binary Phase Shift Keying (BPSK) on one side of the pass band. This paper describes the Fox-1E telemetry including a new feature called Whole Orbit Data. It does not describe the modulation technique itself.

## How Can We Use Six Times The Data Rate?

As the hardware design for the Fox-1E solidified, it was clear that we would be able to send data to the ground at about six times the rate that was available with the FM Foxes. However, the design also envisioned relatively small hardware changes compared the FM Foxes. Thus telemetry data continues to be produced at about the same rate. We had to consider how to make best use of the additional downlink data rate given the limitations on telemetry data generation.

Mike McCann, KB2GHZ, proposed that Fox-1E should have the ability to provide "Whole Orbit Data" (WOD)[1]. In other words, collect satellite health data as well as science experiment data continually, save it in on-board memory, and downlink it in addition to the real-time and min/max data the FM Foxes send.

We decided to use the extra data rate for 3 things:

1. The majority of the additional rate is used to send previously-saved WOD
2. There are a few more data items included in the telemetry
3. We can put more data in a single frame, reducing the overhead.

## Fox-1E Downlink Protocol

The 2017 AMSAT Symposium Proceedings contain a large amount of information about Fox-1E that is not in the form of a traditional paper. Rather there are copies of specifications and schematics that serve to release them into the public domain. Two of them relate to this paper, and I will not duplicate that information:

- *AMSAT Fox-1E Downlink Proposal* written by Chris Thompson, G0KLA – The Physical and Link layer for Fox-1E telemetry

[1] *Fox-1E - Proposed IHU Firmware Features*, Mike McCann, KB2GHZ, AMSAT Symposium Proceedings, 2015.

- *Fox-1E Downlink Spec* written by Burns Fisher, W2BFJ and Chris Thompson, G0KLA – A spreadsheet with the detailed contents of each downlink frame and payload.

In addition, earlier documents that describe the FM Fox telemetry scheme are still partly relevant. In particular, a paper by Phil Karn et al describes the forward error correction and the 8b10b line encoding, which all the Fox satellites use.[2]

## *Packets and Payloads*

In the Fox satellites, telemetry is sent in units called frames or packets. Each frame starts with a header followed by a set of data, followed by check blocks used by the Reed-Solomon forward error correction (FEC). The header specifies the spacecraft time, spacecraft ID, and the type of data within the frame. We call the actual data within frames "payloads." There are 4 different types of payload sent in FM Fox low speed telemetry frames and each frame contains one payload. Each FM Fox payload is 58 bytes and the frame includes as overhead the header, as well as 8b10b line encoding and the FEC check blocks.

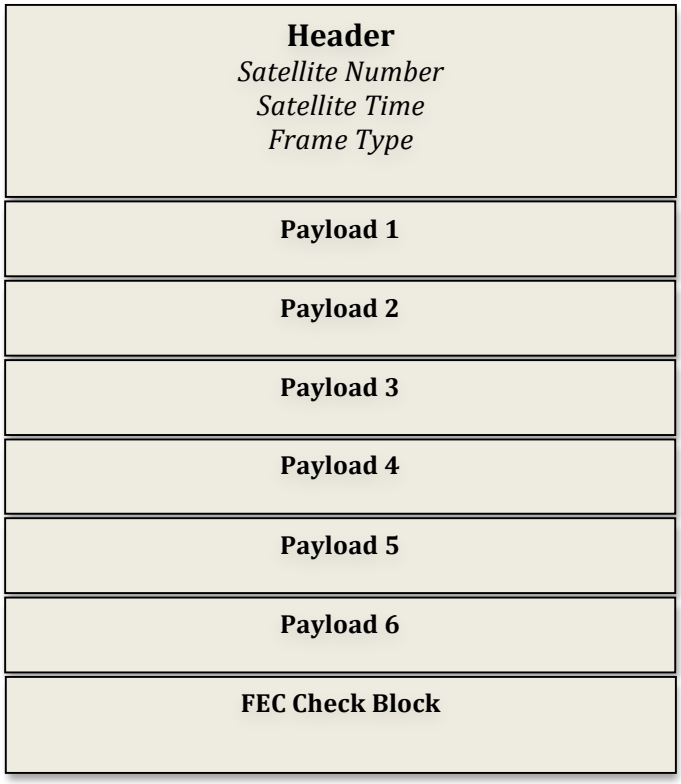



**Figure 1: A Fox-1E Telemetry Frame**

Because of the extra speed in Fox-1E, we can make the payloads a bit longer (78 bytes), and send 6 payloads for each frame.

Fox-1E also adds two different payload types, increasing the number to 6 (unrelated to the 6x data rate, or to 6 payloads per frame!)

The payload types are as follows:

1. Real-time Health Data
2. Minimum Values Health Data
3. Maximum Values Health Data
4. Real-time Science Data
5. Whole Orbit Health Data (Fox-1E only)
6. Whole Orbit Science Data (Fox-1E only)

The Fox-1E Downlink Spec, referenced above, defines the contents of each of these payload types.

Figure one shows a frame, which consists of a header, a sequence of 6 payloads and an FEC Check Block. There are 5 possible sequences of payloads, called Frame Types, numbered 0 through 4. Those types and their payload contents are:

0. All-WOD: Payloads 6, 5, 6, 5, 6, 5

[2] *Coding and Modulation Design for AMSAT Fox-1,* Phil Karn KA9Q, Paul Williamson KB5MU, and Michelle Thompson W5NYV; AMSAT Symposium Proceedings, 2013

1. Real-time: Payloads 6, 5, 6, 5, 1, 4
2. Min-Max: Payloads 6, 5, 6, 5, 2, 3
3. Real-time Beacon: Payloads 1, 5, 5, 5, 5, 5
4. WOD Beacon: Payloads 5, 5, 5, 5, 5, 5

During normal operation, frame types 0, 1, and 2 are sent. In some special modes like Science or Safe Mode, types 3 and 4 are also sent. Notice that EVERY frame type contains at least 4 WOD payloads and All-WOD, type 0, contains 6. All-WOD frames are sent most frequently in normal operation. For each minute of telemetry, we send about 12 frames consisting of 7 All-WOD frames and 5 Real-time or Min-Max. Normally then, we send 62 WOD payloads per minute: 31 Health WOD and 31 Science WOD.

### *Format of Payloads*

The *Fox-1E Downlink Spec* referenced earlier is a spreadsheet that describes the frame header, a data structure called "common" and a separate data structure specific to each type of payload as shown in Figure 2. We use a Python program to read the spreadsheet and output C-language header files for the spacecraft flight software. There is a similar program to build spacecraft description files for the Fox ground software, FoxTelem.

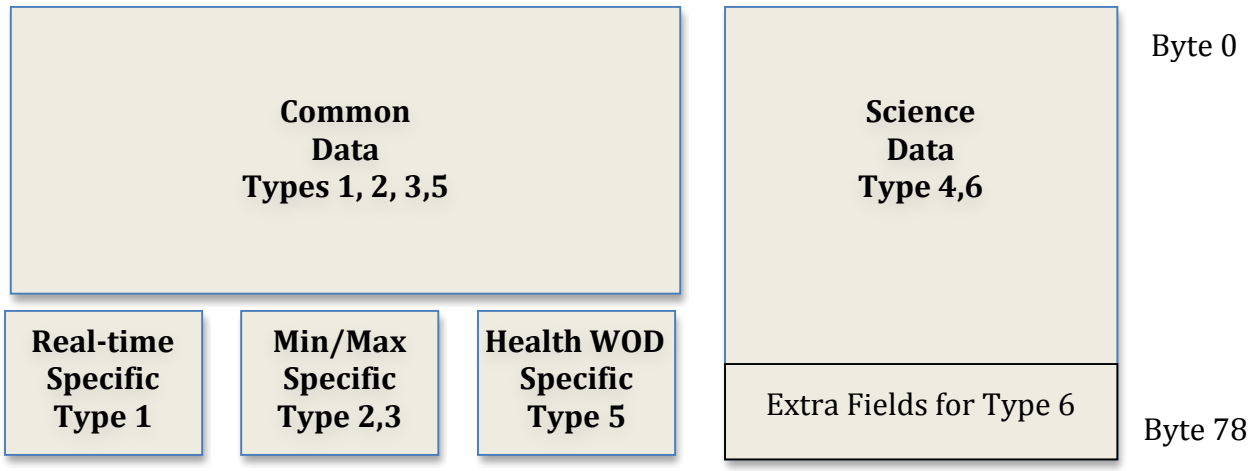



**Figure 2: Payload Formats**

Each health payload (real-time, min, max, and WOD) consists of the common data followed by the payload type-specific data. Common data is the same in every health payload. Payload-specific data includes some status information that does not have to be sent with every health payload, and for min, max, and WOD includes a time stamp indicating either when the most recent min or max change occurred or when the WOD payload was originally collected.

Note that the science data does not use 'common'. It consists completely of data specific to the science experiment.

# How Does WOD Work?

## *MRAM*

All the Fox-1 spacecraft contain a component called MRAM (Magnetoresistive Random Access Memory), a type of non-volatile memory with especially good characteristics for our purposes: These parts have essentially no limit on the number of write cycles possible, and they have good resistance to radiation. In all the Fox-1 spacecraft MRAM is used to store the minimum and maximum health values that the software has seen. In addition, MRAM in the FM Foxes contains the digitized announcements by "The Voice Of Fox", Veronica Monteiro. For Fox-1E, there are no recorded voice announcements, so we used the MRAM space to store Whole Orbit Data.

## *Circular Buffer*

The MRAM space vacated by the FM Fox voice announcements is arranged as a circular or ring buffer. Wikipedia has a nice animated picture of a circular buffer, which I reproduce here (not animated) as Figure 3.

**Figure 3: A Circular Buffer**

Note that this buffer has two pointers: A read pointer and a write pointer. Not surprisingly, software uses the write pointer to enter data and then moves the write pointer forward. Similarly, it reads data at the read pointer location and moves the read pointer forward. Wikipedia describes the most common implementation: if the write pointer laps the read pointer, the buffer is full and the software stops writing and loses data (this is what has happened when Windows beeps at you as you try to type to an unresponsive program). Similarly, if the read pointer reaches the write pointer, there is no more data available and the program stops reading. For WOD we do it slightly differently. We are transmitting WOD data faster than we write it, and we wish to send many copies of the same data. Thus the read and write pointers are completely independent, so that when the fast-moving read pointer encounters the write pointer it just continues on. (The software is careful not to read and write at the same time.) Another way to look at it is that the write pointer is the boundary between the oldest saved data and the newest. Every time there is a new set of data to write, it is written over the oldest data in the ring. Every time the read pointer crosses the write pointer, it starts over again with older data.

## *Getting All the Data to the Ground*

Suppose that the MRAM has room for 124 WOD payloads, WOD data is collected once per minute, and 31 WOD payloads are transmitted per minute. That means that each WOD data has a lifetime in MRAM of 124 minutes before it is overwritten. Since we transmit 31 payloads per minute, we have transmitted the entire buffer in 4 minutes. Thus each payload is transmitted about 124/4 or

31 times. You can see that if the memory changes size, we still send each payload 31 times, but the payloads will be less or more spread out in time.

I said earlier that WOD data contains the spacecraft time at which the data was collected. This time uniquely identifies a WOD payload. The Fox-1 ground software is designed to ignore duplicate payloads no matter how many times they are received.

Despite the satellite not knowing whether what it sends is actually received, we have chosen a ring buffer size so that at least one ground station is likely to receive at least one copy of each WOD record that the spacecraft has collected. We determined this ring buffer size value by simulating WOD telemetry being received by ground stations that currently contribute AO-85 data regularly.[3] After launch, we can optimize the ring buffer size by changing it with a ground command.

## Summary

This paper is one of a set of documents, all appearing in various AMSAT Symposium Proceedings, that describes the format and some of the rationale behind the telemetry on Fox-1E, also known as RadFxSat-2. While the data payloads transmitted by Fox-1E are specific to that satellite, the BPSK modulation technique is standard. This paper and the others referenced here should provide enough information so that an ambitious satellite listener could write his or her own decoder. Of course an even easier way to decode the Fox-1E telemetry is to use AMSAT's FoxTelem program. A version of FoxTelem supporting Fox-1E is currently under test and will be available before the satellite is launched.

[3] *A Whole Orbit Data (WOD) Simulation based on Satellite Orbit Prediction Software*, Carl Wick, D.Sc, N3MIM, AMSAT Symposium Proceedings, 2017.

# Techniques for Interpreting Fox-1 MEMS Gyro Readings

# Or

# Experimenting with an Experiment

*Burns Fisher, W2BFJ*
*w2bfj@amsat.org*

## Abstract

Since the launch of Fox-1A (now AO-85), many of us have been looking at telemetry data coming from the Penn State Erie MEMS Gyro experiment and trying to understand what it means. Unfortunately, between the time of the experiment proposal and Fox-1A's actual launch the students who proposed the experiment have graduated; thus we are on our own.

In this paper, I describe some trigonometric concepts and techniques which we can use to help interpret the data from the MEMS experiment, as well as a ground-based experiment I ran in order to confirm these techniques. Note that all Fox-1 satellites (5 total planned) contain the same MEMS gyros so there will be more data to come.

## What This Paper Is and Is Not

The main intent of this paper is to point out some analytical techniques and to describe some ground-based experiments that I used to validate them. It is NOT a detailed analysis of AO-85 behavior. It is my hope that readers can use these techniques better to analyze Fox-1 satellite behavior. I'm sure papers with further analysis would be welcome in the AMSAT Journal or future Symposia.

## The Fox-1 MEMS Gyros

AMSAT's Fox-1 satellite program was designed from the start to fly science experiments that had been proposed by students and universities. This capability would contribute not only to AMSAT's mission of STEM education, but would also provide leverage to get launch grants (for example from NASA's ELaNa program). One early experiment that we picked was from a group of Penn State Erie students, who proposed to fly some MEMS (MicroElectroMechanical Systems) gyros to determine how they would operate in space. Although the original plan was to fly them on a separate experiment board, our final design put two gyro packages directly on the IHU (Internal Housekeeping Unit—computer). This means that ALL Fox-1 satellites have this experiment on board.

The gyros we are flying are ST Microelectronics LPY410AL dual-axis gyro chips, which report the rotation rate around each of their two measurement axes, X and Z. We mounted two chips on each IHU at right angles to each other. Thus we can get two Z-axis readings as well as one each for X and Y. Their location is about 25mm from the center Z-axis of the IHU board. (They are circled in figure 1 near the hand-written '8').

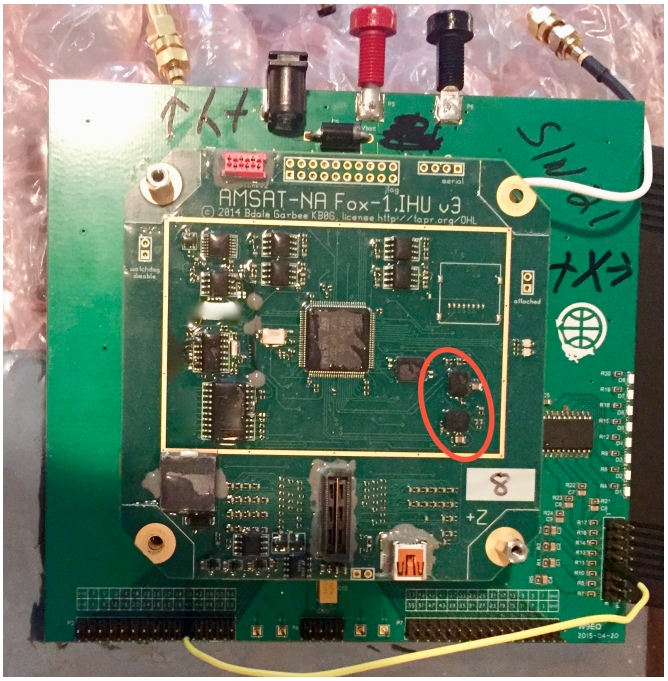



**Figure 1 - A Fox-1 IHU On A Breakout Board With Gyros Circled**

These units provide the rotation rate as analog voltages, which are digitized by the IHU processor's built-in 12-bit Analog to Digital Converter (ADC). Readings from the X-, Y-, and one of the Z-axes are taken approximately 2ms apart. The most recent readings from the gyros are used every 4 seconds to calculate the minimum and maximum value seen, and these min/max data are stored and sent to the ground every minute. In addition, the most recent real-time reading is sent in telemetry about 4 times per minute. Naturally, not all readings that are sent are actually received.

The ADCs have a reference voltage of 3.0V so their 12-bit range of digitized output, 0-4095, should ideally represent a measured voltage range of 0.0-3.0V. In practice, the ADCs are not completely linear; however in this paper and in the calculations I provide, I ignore that non-linearity.

The ADC output is measured from the gyro's "4x" pin which, according to the spec sheet, means that the minimum and maximum rotation speeds that can be measured are -30 degrees per second (DPS) to +30 DPS (-5 to +5 RPM). This rotation speed range should give an output voltage range of about .5V to 2.5V, with 1.5V meaning 'at rest'. However, the spec sheet also states that one should measure the at rest value after the chip is mounted on the board (and the board is mounted in the spacecraft, we assume) to get the true at-rest value, and then calculate the spin using 33mv change per DPS.

We would like to determine parameters like the satellite spin axis and rotation rate around that axis and in the process determine whether the gyro output makes sense at all. To get that information, we need some trigonometry.

## Some Trigonometry

Many hams are familiar with the basic trigonometry functions and how they apply to geometry and electronics. Nonetheless, I will briefly review some concepts so that we are all thinking the same way when I talk about applying them to the MEMS gyros.

First look at Figure 2, a simple X-Y coordinate system with a unit circle (circle of radius 1) and a vector (line with length and direction) drawn in the circle. The angle between the X-axis and the vector is labeled A. The angle between the vector and the Y-axis would be (90-A) degrees.

Now what are the coordinates of the endpoint of the line? The diagram shows that with a vector length of 1, the X-coordinate is the cosine of the angle A and the Y-coordinate is the sine of the angle A.

Next, suppose the length of the vector was NOT actually one but some other number. You are given the x and y coordinates of the endpoint (say x and y). What is the length of the vector? The Pythagorean theorem says that the answer is $\sqrt{x^2 + y^2}$. If it were in 3 dimensions, the answer would be $\sqrt{x^2 + y^2 + z^2}$. (You can double-check this with length 1 also!)

We need a couple more terms that can be extracted from this picture. First suppose we don't know the angle or the sine or cosine of the angle, but we know the vector's endpoint coordinates. We can calculate the "X direction cosine" by dividing the X coordinate by the length of the vector. Similarly, the Y direction cosine is calculated using the Y coordinate (and in 3 dimensions, the same for Z).

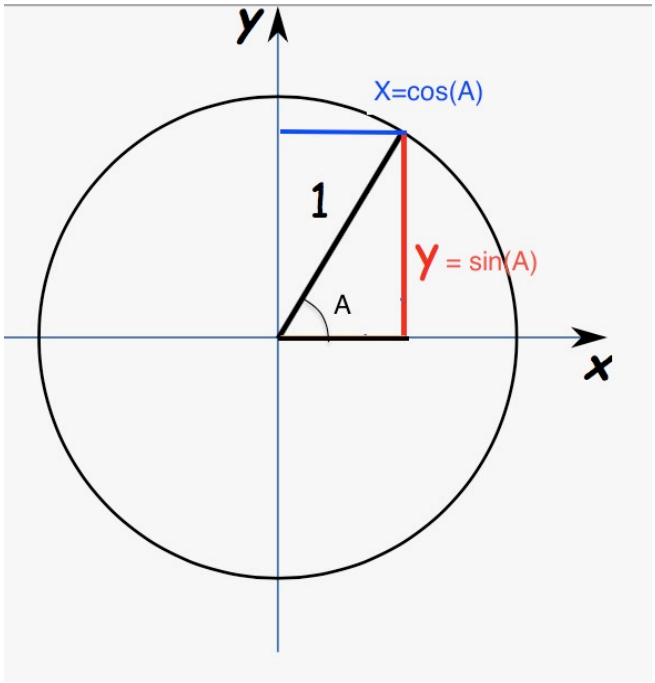



**Figure 2 - A Unit Circle**

And finally, we can calculate the angle A by taking the inverse cosine (also called arccosine) of the x direction cosine. Similarly, we can calculate the angle between the vector and the Y coordinate by using the inverse cosine of the y direction cosine.

Many hams are aware that these geometric concepts apply to radio waves. For example, looking again at figure 2, suppose you have a vertical dipole antenna, which is linearly polarized along the Y-axis. A signal is transmitted through another linearly polarized antenna, but this one is aligned along the vector in the drawing (assume the signal arrives at the same polarization it left with). Then the strength of the signal you receive will be only Cosine (A)*X where X is the strength you would receive if the polarization of transmitter and receiver were the same.

I was pleased when my research discovered that these concepts also apply to MEMS gyros. If a MEMS gyro is lined up on the X- and Y-axes of the picture and is spinning around an axis that matches the vector, then the gyro should read the sine and cosine of the actual rotation rate. Extend this to 3 dimensions and we can calculate both the rotation rate around the spin axis and the angle of the spin axis to each of the X-, Y-, and Z-axes.

## Ground Based Experiments

### *Methodology*

To confirm the predicted behavior described in the last paragraph for at least a couple of rotation axis angles, I used some spare hardware I had in the AMSAT NH Software Lab:

- IHU Serial Number 8 running RadFxSat flight software
- Fox-1 Receiver Board S/N 2 (prototype)

- Fox-1 Transmit Board S/N 2 (prototype)
- Fox-1 Breakout Board
- Kenwood TH-D72 HT set to transmit 67Hz on the Rx uplink frequency
- FunCube Dongle V2
- Windows 7 Personal Computer running FoxTelem
- Leadleds Motorized Display Turntable (nominally 3 or 6 RPM)
- Alkaline D-Cells soldered together to power the satellite boards
- Assorted boxes and static-safe bubble wrap and bags

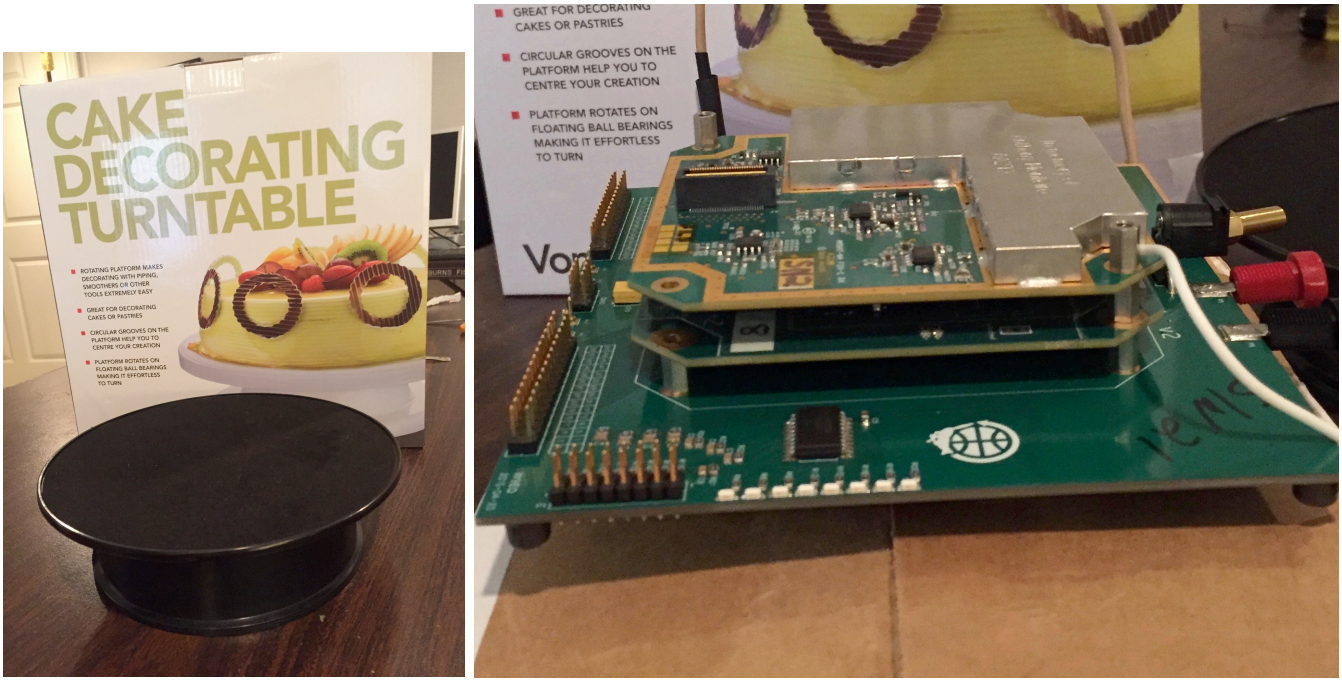



**Figure 3: Turntable and IHU/Rx/Tx Setup (Tx is below the breakout board)**

Figure 3 above shows the components of the experimental setup. I powered the Fox-1 partial stack using the batteries stored in the box below and connected via the breakout board. The box and satellite boards are on the turntable. For each experiment, after starting the turntable, I used the HT to send an uplink signal triggering the transponder. I collected the X-, Y-, and Z-gyro data with FoxTelem along with the satellite time (reset/seconds since reset) and placed these values in a spreadsheet.

I ran several different experiments. Most experiments were running at the turntable's slow speed, which I measured to be about 3.75 RPM, although I did run one experiment at the turntable's high speed of about 7 rpm to understand what happens when the rotation speed exceeds the gyro's spec. I varied the direction of rotation via a switch on the turntable (both

directions were about the same speed), and changed the angle of the axis of rotation by tipping the IHU on the turntable as shown in figure 4.

**Figure 4: Rotation on a non-Orthogonal Axis**

For each experiment, I collected a number of data points. Fox-1 FM satellites send real-time data about every 15 seconds, so the data points are generally spread over the course of a minute or two. It is surprising that the variance among the readings for the same setup is so great. Without further work, I can only guess that the variance comes because the board is not perfectly level, the turntable does not move perfectly evenly, and the ADC is not perfectly consistent for each reading.

For each data sample, I calculated the voltage that the ADC was reporting and based on one of the initial 0-rotation data reports, I then calculated the rotation rate that the gyros were reporting in each axis. Using this data, I calculated the spin rate using the Pythagorean theorem, the direction cosine for each direction, and the angle of the rotation axis from each coordinate axis.

## *Results*

The results at 3.75 RPM approximate "ground truth", although the imprecision noted above carries through to the final values. An abbreviated table of results is shown in Appendix A. For clarity, I have hidden many of the intermediate result columns, including the raw data I collected, and the direction cosine values I calculated. The complete spreadsheet will be available for the foreseeable future at http://burnsfisher.net/AMSAT/Symposium2017/MEMSGyroData.xlsx.

Notice that for experiments listed as Z Plane Flat, i.e. spin axis parallel to the Z-axis, the results show that the calculated angle between the spin axis and the Z-axis is close to 0, while the calculated angle to the X- and Y-axes are both close to 90 degrees, all as expected.

For the section where there was approximately a measured (with an iPhone) angle of 76 degrees to the X-axis, the calculated angle between the spin axis and the X-axis is indeed in the high 70s.

For the section where the measured spin axis angle was about 73 degrees to the Y-axis, the calculated angle was in mid-70s.

Another question that I tried to answer is whether the fact that the gyros are not on the center of the satellite X, Y, and Z-axis is important. There is nothing mentioned in the data sheet about this issue, and I would expect it made little difference (DPS is DPS no matter the center). In order to get a feeling for whether being off-axis made a difference, I placed the gyros approximately 10 cm away from the axis of rotation with their axis parallel to the actual rotation axis. The results show that the error is a bit higher, but still relatively close to 0 degrees in Z and 90 degrees in X and Y.

# What is Next?

## *Using Real Satellite Data*

AO-85 will have been sending MEMS gyro data for over two years by the time this paper is published. This data is easy to acquire using FoxTelem . You can collect your own, or use FoxTelem to download all data from the central server. Then graph the X-axis gyro, add the Y- and Z-axes to the graph and push the "floppy-disk" button to store it as data comma-separated values. Import the data as a CSV file into Excel and apply the same formulas that I used. Now you will be able see the actual rotation rate around the current spin axis, as well as the angle of the spin axis to each coordinate plane of the satellite. One thing to remember is that all of these axes are *relative to the coordinate system of the* ***satellite***, not to the ecliptic or the earth's equator or anything "absolute". For example, we know that due to its passive magnetic stabilization, AO-85 (and the other Fox-1 satellites) should reverse their orientation twice per orbit, roughly as it crosses the equator. Suppose the satellite is spinning on the Z-axis and then flips over 180 degrees. Looking from a fixed location in space we would see the rotation around Z going the opposite direction. However, that will not be the case for the MEMS gyro readings since the Z-axis of gyros themselves has reversed.

## *Questions To Answer*

Starting with the basic techniques in this paper, here are some ideas for extending this work:

- There is some question as to whether prelaunch and launch conditions could damage MEMS devices temporarily or permanently. Is the AO-85 data nonsensical right after launch, and does it improve as time goes on?
- What does the occasional sudden gyration of gyroscope data around the time of the highest temperatures mean? Could the spin rate be exceeding the gyro's capabilities? Could something different be happening that is much more understandable when subjected even to the simple analysis here?
- Is there any way to determine some absolute satellite orientation? Combining MEMS data with solar panel data (which might tell us where the sun is)?
- Is there some way of measuring nutation in the spin axis (a periodic oscillation of the spin axis orientation, such as when you poke the end of a toy gyroscope with your finger)?
- Can we see the satellite reverse its orientation? At what latitudes does this happen?

# Conclusion

I have provided evidence that some simple trigonometric analysis of the MEMS gyros on Fox-1 satellites can provide useful information when a satellite is on the ground. The calculations I used are publically available. The experiments I did are only a starting point for further analysis of the more complex situation when the satellite is in orbit.

# Appendix A

Gyro Calcs

## Gyro Experiments

| | Raw Data | | | | Scaler Rotation | Spin Axis Angle From Coordinate Basis Axes | | |
|---|---|---|---|---|---|---|---|---|
| | Satellite Time | RPM | | | RPM | Degrees | | |
| *Z Plane Flat* | Seconds | Z | Y | Z | | X | Y | Z |
| | | | | | **No Rotation** | | | |
| | *14* | *0.00* | *0.00* | *0.00* | *0.00* | | | |
| | | | | | **CW 3.75 RPM** | | | |
| | 136 | -0.14 | 0.12 | -4.32 | 4.32 | 88.17 | 88.38 | 2.44 |
| | 151 | -0.12 | 0.30 | -4.07 | 4.08 | 88.29 | 85.83 | 4.51 |
| | 165 | 0.08 | 0.24 | -4.44 | 4.44 | 88.95 | 86.91 | 3.27 |
| | 180 | -0.26 | 0.33 | -4.46 | 4.48 | 86.67 | 85.83 | 5.34 |
| | 211 | -0.22 | 0.01 | -3.70 | 3.70 | 86.54 | 89.81 | 3.47 |
| | 226 | -0.02 | 0.20 | -4.79 | 4.79 | 89.81 | 87.66 | 2.34 |
| | 240 | -0.02 | 0.39 | -4.20 | 4.22 | 89.72 | 84.64 | 5.37 |
| | 255 | 0.20 | 0.01 | -5.07 | 5.07 | 87.70 | 89.91 | 2.30 |
| | 284 | 0.24 | 0.60 | -5.09 | 5.13 | 87.37 | 83.27 | 7.24 |
| | 299 | 0.02 | 0.18 | -4.95 | 4.96 | 89.81 | 87.88 | 2.12 |
| | | | | | **CCW 3.75RPM** | | | |
| | 331 | 0.37 | -0.18 | 4.33 | 4.35 | 85.07 | 87.59 | 5.49 |
| | 345 | 0.13 | -0.22 | 3.71 | 3.72 | 87.99 | 86.55 | 3.99 |
| | 360 | 0.24 | -0.09 | 4.08 | 4.09 | 86.69 | 88.69 | 3.56 |
| | 374 | 0.03 | 0.02 | 4.15 | 4.15 | 89.55 | 89.72 | 0.53 |
| | 402 | -0.30 | 0.00 | 3.37 | 3.38 | 84.89 | 90.00 | 5.11 |
| | 416 | 0.48 | -0.48 | 4.20 | 4.25 | 83.46 | 83.52 | 9.22 |
| | 431 | -0.27 | -0.32 | 3.94 | 3.96 | 86.06 | 85.41 | 6.06 |
| *76 Degree Angle On X Axis* | | | | | | | | |
| | | | | | **CW 3.75 RPM** | | | |
| | 694 | 0.91 | 0.17 | -4.31 | 4.41 | 78.13 | 87.78 | 12.08 |
| | 708 | 0.85 | 0.13 | -4.10 | 4.19 | 78.35 | 88.16 | 11.80 |
| | 723 | 0.88 | 0.02 | -4.75 | 4.83 | 79.52 | 89.71 | 10.48 |
| | 737 | 0.92 | 0.03 | -4.37 | 4.47 | 78.09 | 89.64 | 11.92 |
| | 752 | 1.16 | 0.30 | -3.84 | 4.03 | 73.21 | 85.77 | 17.35 |
| | 766 | 0.62 | -0.04 | -3.59 | 3.65 | 80.24 | 89.43 | 9.77 |
| | 781 | 1.03 | 0.25 | -3.89 | 4.03 | 75.21 | 86.41 | 15.24 |
| | | | | | **CCW 3.75 RPM** | | | |
| | 829 | -0.87 | 0.06 | 3.85 | 3.95 | 77.34 | 89.12 | 12.69 |
| | 844 | -0.85 | -0.01 | 4.14 | 4.23 | 78.34 | 89.89 | 11.66 |
| | 859 | -0.73 | 0.06 | 3.99 | 4.06 | 79.67 | 89.14 | 10.37 |

Gyro Calcs

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 873 | -0.58 | 0.02 | 3.48 | 3.53 | 80.58 | 89.74 | 9.42 |
| 893 | -0.77 | -0.15 | 4.33 | 4.40 | 79.95 | 87.99 | 10.25 |
| 908 | -0.89 | 0.00 | 4.08 | 4.18 | 77.64 | 89.94 | 12.36 |
| 923 | -0.70 | -0.05 | 3.94 | 4.00 | 79.87 | 89.24 | 10.16 |

*73 Degrees on Y Axis*

**CW 3.75 RPM**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 186 | -0.23 | -1.50 | -4.74 | 4.98 | 87.38 | 72.41 | 17.79 |
| 201 | -0.11 | -1.13 | -4.18 | 4.33 | 88.60 | 74.94 | 15.13 |
| 215 | -0.07 | -0.82 | -4.06 | 4.14 | 89.04 | 78.57 | 11.47 |
| 230 | 0.02 | -1.05 | -4.29 | 4.41 | 89.74 | 76.19 | 13.81 |
| 244 | -0.08 | -1.14 | -4.54 | 4.68 | 89.01 | 75.88 | 14.15 |
| 259 | -0.09 | -0.87 | -3.68 | 3.79 | 88.65 | 76.77 | 13.30 |
| 274 | -0.05 | -1.26 | -4.99 | 5.15 | 89.46 | 75.88 | 14.13 |
| 288 | -0.19 | -1.29 | -4.60 | 4.78 | 87.76 | 74.32 | 15.85 |
| 303 | -0.10 | -0.86 | -4.38 | 4.46 | 88.70 | 78.87 | 11.21 |
| 317 | -0.10 | -0.50 | -3.14 | 3.18 | 88.24 | 80.89 | 9.28 |

**CCW 3.75 RPM**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 367 | 0.18 | 1.25 | 4.11 | 4.30 | 87.62 | 73.12 | 17.05 |
| 382 | 0.07 | 0.93 | 4.25 | 4.35 | 89.04 | 77.71 | 12.33 |
| 396 | 0.27 | 0.70 | 3.72 | 3.80 | 85.89 | 79.33 | 11.45 |
| 411 | 0.25 | 1.11 | 4.30 | 4.45 | 86.80 | 75.61 | 14.76 |
| 438 | 0.24 | 1.14 | 4.24 | 4.40 | 86.88 | 74.96 | 15.38 |
| 453 | 0.22 | 1.25 | 4.53 | 4.70 | 87.28 | 74.56 | 15.69 |
| 467 | 0.13 | 0.80 | 3.21 | 3.31 | 87.82 | 76.08 | 14.10 |
| 482 | 0.23 | 1.22 | 4.38 | 4.56 | 87.14 | 74.47 | 15.81 |

**CW 3.75 RPM Gyros off center**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 71 | -0.58 | 0.27 | -4.43 | 4.47 | 82.58 | 86.56 | 8.18 |
| 86 | -0.39 | 0.20 | -3.98 | 4.01 | 84.41 | 87.15 | 6.28 |
| 100 | 0.37 | 0.24 | -4.45 | 4.47 | 85.20 | 86.98 | 5.67 |
| 144 | -0.82 | -0.01 | -4.38 | 4.46 | 79.38 | 89.90 | 10.62 |
| 158 | 0.20 | 0.05 | -4.02 | 4.02 | 87.22 | 89.25 | 2.88 |
| 173 | -0.06 | 0.20 | -4.14 | 4.14 | 89.16 | 87.30 | 2.83 |
| 188 | -0.56 | 0.28 | -4.31 | 4.36 | 82.66 | 86.36 | 8.20 |

# Optimally Calculating and Plotting Satellite Subpoints for AO-85 (Fox-1A) CPU Resets

Douglas D. Quagliana, KA2UPW/5
dquagliana@gmail.com

**keywords**: bash, scripts, predict, date, bc, HTML, Google maps, AO-85, Fox-1A, telemetry, CPU, resets, South Atlantic Anomaly

**AO-85, Upsets and Resets; the Early Years**[1]

Before it was even launched, during development of the Fox-1A satellite, Tony Monteiro, AA2TX (SK), predicted that the satellite could reset often, frequently over the South Atlantic Anomaly, and he encouraged the development of an onboard recovery mechanism to survive these resets. Based on our experiences with the ARISSat-1 satellite resetting on every orbit, we knew it would be every desirable to know when each reset occurred and how many resets had occurred. ARISSat-1 had no way of counting the resets or saving the time of the reset. After several days in orbit the ARISSat-1 battery would no longer take a charge, causing the satellite to power off each and every time it entered eclipse and power on/reset every time it exited eclipse.

**No matter where you go, there you are**[6]

For Fox-1a, we have a mechanism to determine the number of resets and the time of the reset. Thanks to the onboard software, the non-volatile MRAM, the telemetry downlink, a worldwide network of telemetry ground stations, the FoxTelem telemetry software, and the AMSAT volunteers, there is an up-to-date master list of all the resets and reset times available on the AMSAT website. Using the satellite reset time, which is the Unix time in milliseconds since 1/1/1970, and the NORAD TLEs for the AO-85 satellite, we can determine the satellite's latitude and longitude at the time it reset. It's just a set of calculations. The more accurately we know the time when the reset happened, and the younger the NORAD TLEs are, the more accurately we know the satellite's position.[2] Therefore, in order to create as accurate a map as possible, it is necessary to use those TLEs for the satellite that are closest to the desired prediction time. Using two year old TLEs will not give accurate results for the satellite subpoint location.[12]

The end result of all of the above work is that there is a complete list of all of the CPU resets and the epoch time of the reset online at

http://www.amsat.org/tlm/ops/FOX1T0.txt

and the format of each line is just the reset number, a comma, and the reset time. For example,

0,1444323370000
1,1444836255000
2,1445891544000

In order to determine where the satellite was at the time of the reset we need to find the set of TLEs closest to this date/time and then use "predict" to calculate the satellite subpoint. However the TLEs use a date format of YYDDD where YY is the year and DDD is the day of the year (up to 365, but sometimes up to 366).

**Stand on giant's shoulders: Don't reinvent the wheel, or calendar calculations, or the date command**

Assuming that we have a collection of TLEs, it is possible to convert the reset time to a date and then search through the collection of TLEs for the TLE that is closest to the reset time. The bash script "find_closest_tle.sh" does exactly this. This bash script accepts a timestamp (the reset time) and then searches through a list of TLEs to find the TLE with the time closest to the reset time. This gives the "freshest" TLEs *for this reset*. To determine the closest TLE, take the year, Julian day and fractional day for the TLE and convert this to a Unix timestamp (seconds since 1/1/1970). Performing date calculations can be non-trivial. Fortunately, we can use the Linux "date" and "bc" commands to perform most of the date calculations without all that tedious mucking about with leap years.[11] The conversion between Unix epoch and YYDDD format is done using the standard Linux "date" command with the "+%s" option. Since the date command has already been written and debugged, and since it already knows about things like leap years, we can leverage the efforts of its author and minimize our own efforts in coding and debugging which could introduce date computation errors. Once the dates is converted, we can compare the TLE timestamp to the desire timestamp and save the TLEs if they are the best seen so far.

**Don't reinvent satellite orbital calculations either, use something that is known to work correctly**

Since we really want to do this for each and every CPU reset, we can use another bash script (convert_T0_time_to_SSP.sh). This script gets the latest list of resets from the AMSAT website using the "wget" command and calls find_closest_tle.sh for each reset. The reset times and best TLEs are passed to the KD2BD predict program which calculates the satellite subpoint latitude and longitude for that time using those TLEs. After we perform the calculations for satellite subpoint for every reset and then we have a list of resets, latitudes, and longitudes. Now we can plot all of these points on a map using Google maps.

**Map Making and the South Atlantic: Here Be Dragons (and Anomalies, and single event upsets, and plastic…lots of plastic)[8]**

There's more than one way to make a webpage with a map using the Google APIs. This is just one of the ways, but it is pretty simple. There is some canned HTML "header" code that needs to be at the top of the webpage, some other HTML "trailer" that needs to be at the bottom, and the list of latitude and longitude points that are placed in between those two sections. When the three pieces are put together

into an HTML file, it creates a webpage with a map. This makes it easy to generate a simple map from a bash script which will answer our question “Where are the CPU resets happening?” This method has the disadvantage that it can only be used to display about four hundred points on the map. Even though there have been over seven hundred resets, many of them are “duplicates” and do not need to be plotted.

This is the HTML “header” code:

```
<html><head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script>
    google.load('visualization', '1', { 'packages': ['map'] });
    google.setOnLoadCallback(drawMap);
     function drawMap() {
       var data = google.visualization.arrayToDataTable([
         ['Lat', 'Long', 'Reset#'],
```

That’s all there is! Then we add our list of latitude/longitude points within square brackets and with a comment containing the reset number and latitude and longitude. The comment will be visible when the user hovers the mouse over the point on the map.

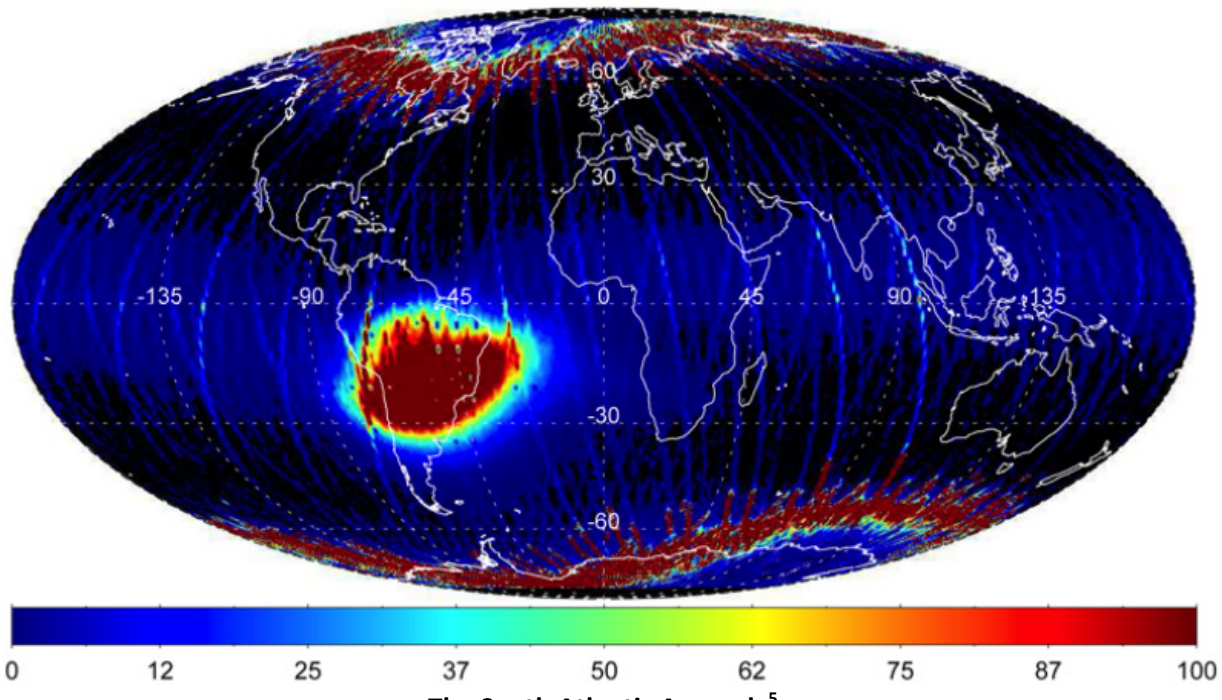
```
[ -56, -24, 'Reset#0 -56, -24' ],
[ 55, 158, 'Reset#1 55, 158' ],
[ 39, 51, 'Reset#2 39, 51' ],
```

… and so on until all of the points are listed. Lastly, this is the HTML “trailer” code:

```
]);
    var options = { showTip: true };
    var map = new
google.visualization.Map(document.getElementById('chart_div'));
    map.draw(data, options);
  };
  </script></head><body><div id="chart_div"></div></body></html>
```

These scripts were written just to “get the job done” with no attempts at optimization. After all, “premature optimization is the root of all evil (or at least most of it) in programming.”[10]

**Figure 1 – The map of satellite subpoints for AO-85 CPU reset numbers 0 through 713.**
**There is an interactive version of this map online at http://qsl.net/ka2upw/ao85_reset_map.html but eventually I would like to get the map and scripts onto the AMSAT website**




**The South Atlantic Anomaly[5]**

**Sticking pins in a map – well, what's the point?**

A quick look at the map of the subpoints for CPU resets shows good agreement with the South Atlantic Anomaly (SAA).[4] It was expected that many of the resets, but not all, would be near or over the SAA and this is indeed what we find. A further analysis of the telemetry will let us look at the type of reset and match those up with the satellite subpoints. It will also be interesting to try and correlate the results of the radiation experiment, time of day, and solar activity with the satellite's resets. If you're interested in analyzing this, please contact the author.

**Credits**

"It is amazing what you can accomplish if you do not care who gets the credit."

--Harry S Truman

I would like to thank the following people who helped make this paper possible.

- Alan Biddle, WA4SCA for keeping track of CPU resets as they happen and sending emails to the FoxTLM mailing list
- Chris Thompson G0KLA/AC2CZ for FoxTelem and the AMSAT telemetry web pages
- Mark Hammond, N8MH for constructive feedback and support
- John A. Magliacane, KD2BD for writing predict and making it open source
- Philip A. Nelson for the "bc" command
- David MacKenzie, for the Linux "date" command and all its options
- Brian Fox and Chet Ramset for bash

And all of the ground stations that sent in telemetry, because without all of the collected telemetry none of this would be possible.

The author encourages and welcomes feedback and commentary, especially constructive criticisms and particularly welcomes deconstructive comments and puns, copies of your satellite recordings and telemetry, and perhaps any suggestions and remedies you might have for excessive "and" conjunctionitis.

Listing 1 - convert_t0_times_to_SSPs.sh

```
#!/bin/bash

# get the latest copy of all of the T0 reset times
# assumes that we have an internet connection
rm FOX1T0.txt 2> /dev/null
wget http://www.amsat.org/tlm/ops/FOX1T0.txt

previous_reset_time=0

# use "for p in $(<filename);" notation .
# If you use the "while read p;" version then the
# last line of FOX1T0.txt will not get processed.
#
for p in $(<FOX1T0.txt); do

   # file format is reset_number, reset_time -- for example
   # 0,1444323370000

   # get just the reset number
   reset_number=`echo "$p" | cut --delimiter=, -f1`

   # get just the reset time, dropping the trailing 000 (milliseconds)
   # leaving just the seconds since 1/1/1970 (Unix epoch time)
   reset_time=`echo "$p"   | cut --delimiter=, -f2 | cut --bytes=1-10`

   if [ "$previous_reset_time" != "$reset_time" ]; then
      previous_reset_time=$reset_time
      ./find_closest_tle.sh  $reset_time > reset_$reset_number.tle
      echo -n $reset_number, > reset_$reset_number.out
      predict -q /home/pi/predict/ka2upw.qth  -t reset_$reset_number.tle  -f AO-85
$reset_time $reset_time >> reset_$reset_number.out
   fi
   echo " Done"
done
```

Listing 2 - find_closest_tle.sh

```
#!/bin/bash
# find closest TLE (to a give date).
# Given a reference date from the user (on the command line), and a file of TLEs
# (all for the same satellite) search the file containing all the TLEs of this
# satellite and find the TLE with the date closest to the reference date.
# The TLE file is assumed to be triple lines of SATNAME, TLE1, TLE2 but not
# necessarily in date order. The TLE file is assumed to contain only TLEs for
# this satellite.
# Background: orbital prediction errors get larger when older TLEs are used.
# In order to get the most accurate predictions we need to use the TLEs that
# were generated closest to the time of the desired prediction.
# Since there could be gaps in the available TLE entries, we can't just take
# the most recent before or most recent after (consider the case where you
# want a  prediction for June 1 and you have TLEs for May 30 then a gap until
# December, or you want June 1 and have TLEs for January then nothing until
# June 4).  We want the closest date.
# Note: requires the "bc" package to do some of the calculations.
#

# Reset times from AO-85 are in milliseconds, so drop the
# trailing 000 to get seconds since 1/1/1970
time_to_match=$1
closest_matching_tle_delta=999999
smallest_delta_time=99999999

# N.B. date command's "j" is day of year 1..366
julian_date_to_match=`date -d @$1 +%j`

# read lines from AO-85.TLE - a single TLE is actually three lines long.
# The first line has the satellite name.  Lines 2 and 3 have the orbital elements.
# For example:
#
# AO-85
# 1 40967U 15058D   15344.32453720  .00001845  00000-0  20662-3 0 00659
# 2 40967 064.7773 101.4283 0217658 263.0133 094.6183 14.74493538009130
#
# 123456789111111111122222222223333333333444444444455555555556666666666
#          012345678901234567890123456789012345678901234567890123456789
# See also:
# https://en.wikipedia.org/wiki/Two-line_element_set
#

# 86400 = 60 seconds per minute times 60 minutes per hour times 24 hours per day
seconds_per_day=86400

line_number_of_element_data=0
IFS=''
while read p; do
   line_number_of_element_data=${p:0:1} # 1 or 2

   if [ "$line_number_of_element_data" != "1" ]; then
      if [ "$line_number_of_element_data" != "2" ]; then
         tle1=$p
      fi
   fi
```

```
   # is this the first line of orbital elements?  (starts with a 1)
   if [ "$line_number_of_element_data" == "1" ]; then
      tle2=$p

      # satellite_number=`echo "$p" | cut --bytes=3-7`          #
      # classification=`echo "$p" | cut --bytes=8`
      # classification: a U here means unclassified.
      # Isn't it odd that all of the satellites are unclassified?

      epoch_year=${p:18:2}
      epoch_whole_days=${p:20:3}
      epoch_fractional_days=${p:23:9}
   fi

   # is this the second line of orbital elements?  (starts with a 2)
   if [ "$line_number_of_element_data" == "2" ]; then
      tle3=$p
      tle_epoch_as_unix_epoch=`date -d "20$epoch_year-01-01 +$epoch_whole_days days -1
day" "+%s"`

      delta_time=$(echo "($tle_epoch_as_unix_epoch + $epoch_fractional_days *
$seconds_per_day) - $time_to_match" | bc -l)

      # kludge to compute absolute value; sqrt(x*x) might be better?
      abs_delta_time=$(echo $delta_time | tr -d -)

      if (( $(echo "$smallest_delta_time > $abs_delta_time" |bc -l) )); then
         smallest_delta_time=$abs_delta_time
         smallest_tle1="$tle1"
         smallest_tle2="$tle2"
         smallest_tle3="$tle3"
      fi
   fi
done < AO-85.TLE
echo $smallest_tle1
echo $smallest_tle2
echo $smallest_tle3
```

Listing 3 - make_resets_map_from_out_files.sh

```
#!/bin/bash
# each of the *.out files is a single line output from "predict"
# prepended with the reset number. Each one looks like this line
# 619,1460541153 Wed 13Apr16 09:52:33  -28  158  245  -27   74   6941   1371 *
# reset,Unix epoch DOW Date Time(UTC)

# PREDICT produces an output consisting of the date/time in Unix format, the
# date and time in ASCII (UTC), the elevation of the satellite in degrees,
# the azimuth of the  satellite  in  degrees,  the orbital phase (modulo 256),
# the latitude (N) and longitude (W) of the satellite's sub-satellite point,
# the slant range to the satellite (in kilometers), the orbit number,
# and the spacecraft's  sunlight  visibility  information.

rm reset_positions.txt 2> /dev/null
# start building HTML table
echo "<table style="width:100%">"  >> reset_positions.txt
echo "<tr>"                        >> reset_positions.txt
echo "   <th>Reset #</th>"         >> reset_positions.txt
echo "   <th>Epoch</th>"           >> reset_positions.txt
echo "   <th>Date</th>"            >> reset_positions.txt
echo "</tr>"                       >> reset_positions.txt

cp map_header.html reset_map.html

for f in `ls *.out | sort -V`
do
   echo -n "Processing $f"
   SatResetNumber=`cat $f | tr -s ' ' | cut -d, -f 1`
   SatSSPLat=`cat $f | tr -s ' ' | cut -d ' ' -f 8`
   SatSSPWLon=`cat $f | tr -s ' ' | cut -d ' ' -f 9`  # west longitude

   # predict outputs west longitude values but google maps uses only
   # latitude values from -180..0..+180 so we need to convert
   SatSSPLon=`echo "define convert ( lon ) { if ( lon < 180.0 ) return ( lon * -1 )
else return ( 360 - lon ) } convert( $SatSSPWLon )" | bc -l`

   echo " Lat=$SatSSPLat WLon=$SatSSPWLon Lon=$SatSSPLon"
   echo "[ $SatSSPLat, $SatSSPLon, 'Reset#$SatResetNumber $SatSSPLat, $SatSSPLon' ],"
>> reset_map.html
   echo "<tr>"  >> reset_positions.txt
   echo "<td>"  >> reset_positions.txt
   cat $f       >> reset_positions.txt
   echo "</td>" >> reset_positions.txt
   echo "</tr>" >> reset_positions.txt
done
echo "</table>" >> reset_positions.txt

cat map_trailer.html >> reset_map.html
cat reset_positions.txt >> reset_map.html

# Append date this map was created so we can verify auto-updates
echo -n "Last update:" >> reset_map.html
date >> reset_map.html
```

Footnotes

1 This subtitle was suggested by Alan Biddle, WA4SCA.

2 Werner Heinsberg would have hated this.

3 Deleted

4 The South Atlantic Anomaly isn't actually centered over the southern portion of the Atlantic Ocean. In fact, the anomaly moves. For more details on the movement of the SAA and a discussion of some special comments on the literature of the SAA, see footnote 5.

5 Stassinopoulos, Epaminondas G., Xapsos, Michael A., Stauffer, Craig A. Forty-Year "Drift" and Change of the SAA. NASA Goddard Space Flight Center Technical Report NASA/TM-2015-217547, GSFC-E-DAA-TN28435. December 1, 2015. https://ntrs.nasa.gov/search.jsp?R=20160003393

6 Weller, Peter, actor. The Adventures of Buckaroo Banzai Across the 8th Dimension. Sherwood Productions. 1984.

7 I was motivated to not re-invent the wheel and use the Linux "date" command for all of the date calculation after reading some of James Miller's (G3RUH) papers including "30.6 Days Hath September" and "Sun's Up". For a collection of Miller's work see http://www.amsat.org/articles/g3ruh/bibliog.txt.

8 Actually, I'm told that there really aren't any dragons at all[9], and in fact, there are very few maps where the phrase "Here be dragons" actually appears despite there being many maps with pictures of dragons and lots of people being familiar with the phrase. I'm assured, however, that there certainly is a lot of plastic in the oceans.

9 Except in Komodo, where there really are dragons, but there are far enough away from the south Atlantic that we can ignore them for purposes of our calculations here. But I digress…

10 This quote is frequently attributed to Knuth, Donald. Computer Programming as an Art. 1974. However, it is also attributed by Knuth to Sir Tony Hoare. However, wikiquotes notes that the attribution to Hoare is doubtful. https://en.wikiquote.org/wiki/C._A._R._Hoare

11 Apologies to "other Doug".

12 Calculating precisely how inaccurate the two year old TLEs would be is left as an exercise to the reader. Hint: use KD2BD's predict program running under Linux.

# A Whole Orbit Data (WOD) Simulation based on Satellite Orbit Prediction Software

Carl E. Wick, D.Sc., N3MIM

## Abstract.

Most satellites, including AMSAT Fox satellites, collect important satellite state and experimental data at periodic intervals throughout an orbit. A complete orbital history of satellite data can be extremely important in not only determining the continuing health of a satellite, but also in determining reasons and corrective action when anomalies are observed. Acquiring and maintaining historical data implies that there must be a system to not only collect sufficient and timely data at the satellite, but it must also ensure that a complete record of this data is received at control points on Earth. In this paper I discuss a simulation method that was used to help answer some important satellite systems design questions concerning the collection, retention, transmission and ultimate reception of "whole orbit data" at an Earth control point.

## Introduction.

Except in very rare instances, once a satellite is launched the only possible view of its current state of operation is through specific health-related data periodically sent earthward. Though each single data set provides valuable information, it is only through complete histories of the data (so-called "whole orbit data") that trends may be noted or subtle anomalies may be discovered. Therefore, in the design of satellites that record and transmit internal system data, it is important to consider a more complete picture; from generation of the data, to queuing on the satellite, to transmission schemes, to reception by ground stations, to final collation of all data at a control point. Because a satellite is typically not recoverable to analyze, rebuild and relaunch, it is important to consider all collection, transmission and re-assembly factors as completely as possible at design time.

When designing whole orbit data systems, it is very desirable to be able to precisely model and predict in advance what will happen as a satellite collects data and then sends the data to ground stations while it orbits the earth. Equally important is to have a model that is as faithful to true operating conditions as possible and one that makes minimal assumptions. Given the remote nature of space, such prediction can often be performed only through carefully constructed simulations.

One type of simulation in common use that accurately depicts the position of a satellite with respect to the Earth is an orbit prediction model, commonly called a satellite tracker. Such a

model can not only predict the location of a satellite in real time, but can also project a satellite orbit in time to show future satellite positions. When the location of a satellite in an orbit is known, that location can then be used to determine which portion of the Earth could receive the data, how much data could be received at a given transmission rate, etc. Predicted positions will also reveal portions of an orbit in which no reception is likely, for example when the transmission footprint of a satellite is over an ocean or over a relatively unpopulated region. Such knowledge will help in deciding how much data a satellite must retain during an orbit to have reasonable assurance of all orbit data would be received somewhere on Earth.

The simulation described in this paper has at its base a standard orbit prediction program. The code was modified by the author to not only track a collection of satellites, but also to maintain a large database of earth stations that routinely provide satellite telemetry data to AMSAT. The program can determine satellite viewing parameters at each station and use this information to determine the likelihood that data from a satellite would be received. By knowing orbital satellite positions and ground station locations various data assembly and transmission schemes at a satellite can be devised, tested and tuned in an iterative fashion to produce well-functioning design.

## Simulation.

The orbit simulation program used in this study began from one constructed by the author several years prior as a way of answering similar types of questions in the design of a commercial communications satellite system. The orbital prediction portion of this simulation is based on a standard NORAD low-earth orbit-prediction model (NORAD SGP4). For those interested, details of the SGP4 model, including FORTRAN code, may be found at: https://www.celestrak.com/NORAD/documentation/spacetrk.pdf. The SGP4 Fortran code was translated by the author to equivalent C++ and was then placed in a C++ Builder graphics shell. The addition of a graphics shell allows projected satellite positons and ground stations to be viewed on a three-dimensional near-earth representation and it allows for controlling variables from a graphics screen. In normal operation satellite positions are tracked in real-time at a one-second rate, but projecting future satellite positions can be made at substantially higher rates, requiring only about one second per orbit to calculate all ephemeris and communications parameters. Orbit prediction of operational satellites relies on data from standard satellite orbital element sets, which are easily found on the internet. Correct operation of the orbital portion of the simulation was verified by comparing operational satellite positions with other existing prediction models and by verifying with beacon signals from operational satellites.

A substantial earth station model was added to the satellite orbit prediction simulation. In this addition, at each orbit point the earth footprint (area satellite is visible from earth) of the satellite in question is calculated. A database of ground stations is then queried to determine

which ground stations fall in the footprint and at what azimuth and elevation the satellite is seen by each. This database was populated by a collection of ground stations that regularly contribute telemetry to the AMSAT server. Positions of all ground stations in the database are shown on the 3-D earth graphic during operation.

Further modification to the simulation is code to simulate, on a second-to-second basis, data generation, queuing and transmission from a satellite like AO-85 or future Fox satellites to the collection of earth stations. Probability of reception of any one station is determined through "Monte Carlo" statistics based on observed satellite elevation. As a particular simulation progresses, several files are generated to record the progress of a satellite's orbit and to record other important factors such as number of ground stations that see the satellite, amount of data is satellite queues, how much data has been successfully received by ground stations, etc. These data are saved in such a form that they can be viewed in other data viewing and processing programs. I used MATLAB for further processing and data viewing.

Once this software tool was established, I then used it to answer some important design questions related to future AMSAT satellites.

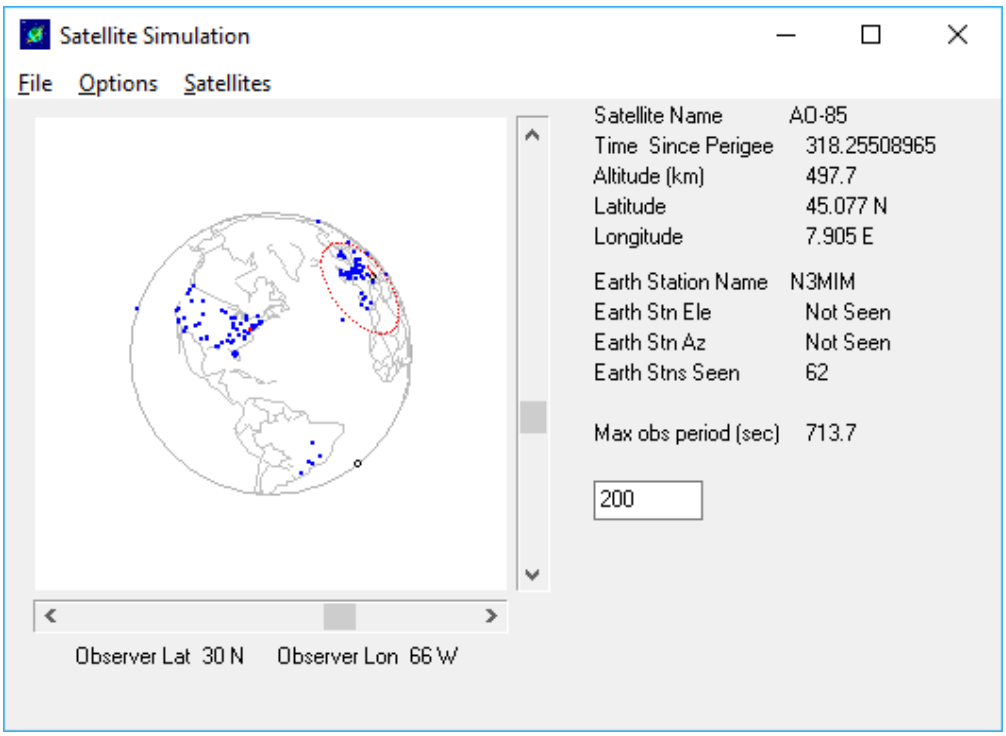



**Figure 1 .** Snapshot of the subject satellite prediction program in operation. The red circular footprint of the satellite of interest (AO-85) may be seen over the European continent. Note the simulation indicates that satellite can be seen by 62 AMSAT contributing earth stations there (blue dots). The program can simultaneously track multiple satellites; a secondary (no footprint shown) fictitious polar orbiting satellite may be seen as a small circle on the earth limb just off South America. Also shown here

is an edit box shown on the lower right, which allows satellite queue size to be changed in simulation runs (current size is 200). In this way parameters can be optimized.

## Some design questions and simulation answers.

Even from the limited view shown in Figure 1 several observations may be made. The first is that the collection of AMSAT contributing earth ground stations are highly localized to a relatively few geographic areas. In the view, America and Western Europe have a heavy concentration of receiving stations, a few receiving stations are located along eastern South America and almost none elsewhere. It becomes obvious that during a satellite orbit there will be times that data passed to the ground will have high probability of reception because many stations are under the footprint, but also during an orbit there may be long stretches where few or no receiving stations are available. This leads to the first question.

**Question 1.** In the likely orbit path of a satellite like AO-85, much of an orbit will the satellite be out of range of any telemetry receiving station? The answer can be valuable in determining how much data must be retained at a satellite in a queue so that a complete history might be received during the course of an orbit or several orbits.

Such a question might be answered in part by projecting a satellite over an orbit similar to AO-85 for a long period of time (one day was chosen in this study). During the projection of the satellite track, each earth station is queried each second to see if it falls in the footprint. Counters are then incremented to show periods of time that no earth station is under the footprint of the satellite, and other counters are incremented to show when at least one station is under the footprint. Since AMSAT satellites (and most cubesats) do not intrinsically know what land mass they are operating over at any time, the data they generate must be held in some type of a queue. This queue must be of sufficient size so that when data is sent from the queue on a continuing basis all data may eventually be seen by a collection of ground stations. It should be obvious that the size of this queue is directly related to the maximum length of time the satellite is without a suitable receiving station under its footprint. Some results of a simulation of this type can be seen in Figure 2.

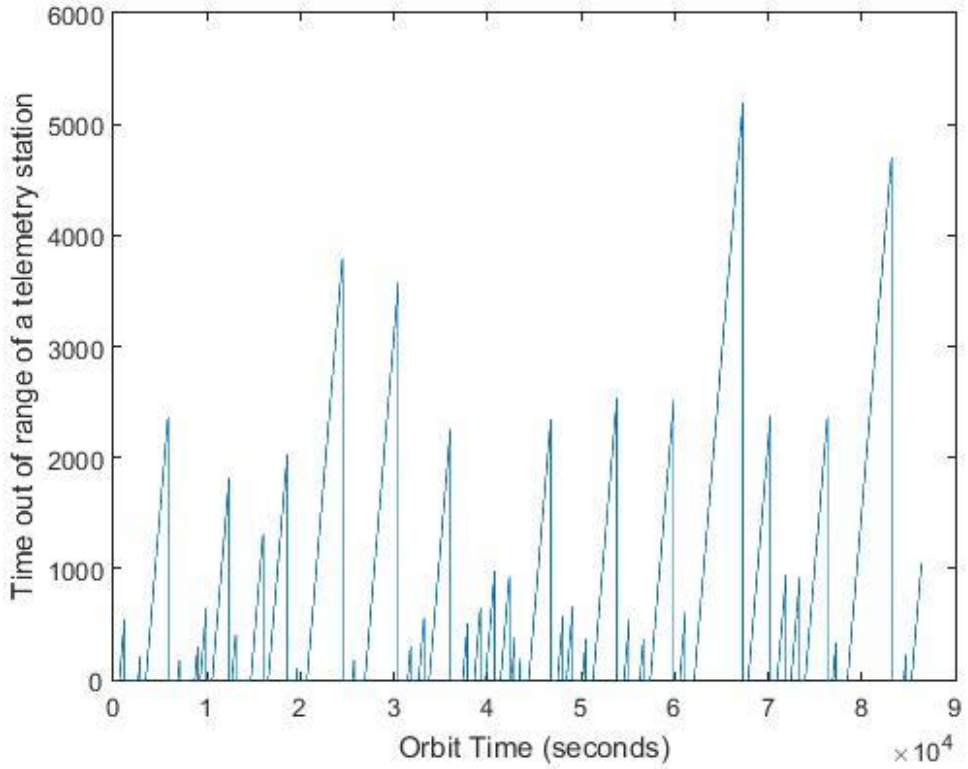



Figure 2 Results of a simulation of an AO-85 orbit over existing AMSAT telemetry receiving ground stations.

In Figure 2 it can be seen that a satellite in an inclined orbit like that of AO-85 sees significant periods of time during which the satellite is not in range of any of the database telemetry receiving sites. The longest period during the 24 hours simulated for an orbit like AO-85 is over 5100 seconds (85 minutes). Other outage times of 2000 and more seconds are very common. If a complete record of data is to be received at the ground at some point, then a satellite data queue would have to hold at least enough data to cover the largest outage period it might encounter, so that it might be able to send the total collection of historic data to the ground piecewise at a later time. It may also be interesting to note that the results of this type of simulation are slightly dependent on orbital plane. For example, an inclined orbit northwest to southeast (like AO-85) can encounter a significant stretch of ocean area, particularly in the Pacific Ocean due to the distribution of land masses, where polar orbits may result in shorter stretches of ocean. Whether this orbit-dependent factor is important also has to do with the geographic concentration of ground stations around the world.

**Question 2.** Can periodic data be held and retransmitted to available ground stations such that all data can eventually be recovered on the ground? That is, can at least one ground station recover at least one sample of every data set generated by the satellite? This is a complicated question that has several variables, which include: how often data is generated by the satellite, the size of data packets to be sent, the data queuing scheme used, the rate data can be transmitted by the satellite to earth, the number of stations under a footprint, and the probability that a viewing ground station can successfully receive a complete data packet.

To answer this question a separate model of proposed satellite queuing strategy was programmed into the simulation as a subroutine. In this queuing strategy a satellite pushes data packets of a known size into an onboard satellite queue at a predetermined sample rate (2 packets per minute in this study). The satellite then sends data blindly to earth round-robin from the queue at a rate that can be supported by the modulation method used for telemetry and the physical size of the packets to be sent (1200 baud and approximately one data packet per second in this study). The satellite must continually send packets in circular buffer style because it has no knowledge of when it is over stations that may receive the packets. It must rely on having a queue that is large enough and transmission rates are fast enough that all the data will be received by some station somewhere. Imperfect ground reception was accommodated by using a probability distribution based on satellite elevation at the time packets are sent from the satellite. This distribution function (0 to 1) assigns low probability when the satellite is near the horizon and high probability if a satellite is near zenith from a receiving station. Monte Carlo techniques are used to determine if a message makes it from satellite to ground station. A random number between 0 and 1 is chosen by the computer for each station falling under the satellite footprint. If the random number is less than the distribution value for the satellite elevation seen, reception is said to have occurred, otherwise reception does not occur. If reception happens at any station for a packet, then the item in the satellite queue is marked as “sent”. As the simulation runs, new data packets are continuously pushed onto the queue at a low sample rate. Once a packet has reached the length of the queue it is considered “lost” unless is has been received at some point by some ground station and marked as “sent”. In a simulation run the queue is checked to see if more un-received packets remain than the projected length of the queue. If this happens, data is presumed lost. By running simulations and adjusting queue sizes to reduce or eliminate lost data a more-or-less optimum size can be chosen for a particular queuing and communications strategy.

From simulations done to answer questions about queue size, an interesting observation emerged. If queue sizes are too small, data will be routinely lost because not enough data is accumulated over long areas of outage and leave the queue before it can be retransmitted and received by a ground station in a populated area. But large queue sizes also lost data - surprise. In this case the problem is that the available ground stations are too concentrated in relatively

small geographic areas and there is simply not enough time during a pass to get substantially through a large queue transmitting data in a circular fashion (at medium data rates) . Essentially the satellite has moved beyond reception range of any ground station before it can get completely around a large queue.

## Summary.

This paper has outlined how a low-earth-orbit tracking simulation was modified and used to begin to answer some basic satellite whole orbit data system design questions. These simulations are based on actual orbital positions taken from existing or proposed orbital element sets and they are based on the actual locations of contributing ground stations. Furthermore, the resulting simulation duplicates the communications functions of a target satellite design and uses actual ground station locations to predict the likelihood that various systems parameters and schemes will or will not work in an operational environment.

# An Update on Past and Future Projects from AMSAT-UK.

David Bowman G0MRF

**Abstract:**

Since 2013, AMSAT-UK, together with AMSAT-NL, and partners from industry and education, have produced 4 CubeSats containing amateur radio transponders. This paper summarises the current status and lessons learned from the AO-73, UKube, EO79 and EO-88 missions. It also looks ahead to the launch of other satellites currently under construction, including the long awaited L/V (1263 to 145MHz) FM transponder on the European Space Agency ESEO satellite.

## AO-73 / FUNcube 1

1U CubeSat with UHF to VHF linear transponder. Telemetry at 1200BPS BPSK at 300mW / 35mW 145.935MHz
Launch: Dnepr from Yasny Russia. 3rd November 2013
Orbit: Sun synchronous orbit at 630km
Collaboration: AMSAT-UK, AMSAT-NL, Innovative Solutions In Space (ISIS bv) Delft, Netherlands

## Current Status

AO-73 has completed over 21,000 orbits and will be celebrating the 4th anniversary of its launch in a few days.
The 2m transmitter runs 'high power' telemetry when AO-73 is in sunlight, switching to transponder + low power telemetry when the satellite is in eclipse. At weekends the satellite's transponder is continuously active from Friday evening to Sunday evening UTC.
As of 1st October 2017, AO-73 has returned more than 5.4 million frames or 1320 MB of unique data via the Data Warehouse. This includes 80.2% of all whole orbit data and 8.9% of real time data (transmitted only once every 5 seconds). During the weekends over Europe, AO-73 continues to attract many QSOs on its transponder.
One 'lesson learned' from AO-73 is that care must be taken with the oscillator design, particularly on a 1U CubeSat. The low mass of the 1U format produces large temperature swings, particularly in eclipse. Of the 3 identical transponders fitted to AO-73 (1U) UKube-1 (3U) and EO-79 (2U), only AO-73 exhibits noticeable drift on the 70cm uplink receiver. The local oscillator is a design that consumes very little DC power, and AO-73 runs several degrees cooler than anticipated during pre-launch analysis. Neither characteristic aids oscillator stability.
The satellite's systems have been very reliable with just 4 computer resets in 4 years. The power budget is excellent and with just 5 to 6% depth of discharge in eclipse, the Lithium Ion battery appears to be in good health despite 21,000 cycles of sunlight and eclipse. At end of battery life, AO-73 has been designed to operate directly from its solar panels. Let's hope it's several more years before we discover if that design feature is successful or not.

## UKube-1 (FUNcube 2)

3U CubeSat with UHF to VHF transponder. + AMSAT telemetry at 1200bps BPSK on 145.915 MHz
Launch: Soyuz from Baikonur in Kazakhstan. 8th July 2014
Orbit: Sun Synchronous at 615km
Collaboration: UK Space Agency, Clyde Space, AMSAT-UK, AMSAT-NL and other payload suppliers.
UKube-1 was the first mission for UK Space Agency. Led by the Scottish company Clyde Space, this 3U satellite functioned as a technology demonstrator for 4 major science payloads. These included a Camera design from the Open University and an experimental random number generator driven by radiation. While data was downloaded from all of the core payloads, the satellite suffered from multiple computer resets per orbit.

In October 2015, after 1 year of operation, The UK Space Agency declared 'end of mission'

**Figure 1 UKube. 3U CubeSat with deployable solar panels at ClydeSpace 2013**

**Current status**

Personnel operating the UKube command station were reassigned to other projects in late 2015. However, the satellite is still active with occasional signals from its primary communications system on 145.840 MHz and with the AMSAT-UK supplied transponder and Tlm subsystem operating 24/7. Although this satellite is not actively managed, it has a large number of solar cells fixed to the 3U structure and on its deployable panels. Consequently the power budget is positive and the Clyde Space EPS and Lithium Polymer batteries have a good state of charge throughout the orbit. The frequency stability of the transponder is better than AO-73 as the mass of a 3U CubeSat reduces extreme swings of temperature. However, the transponder does suffer from desense to its UHF receiver when beacon transmissions occur from its primary communications system. Use of telemetry, in particular Whole Orbit Data from the satellite for educational outreach, has been compromised by the large number of computer resets in the main OBDH which affect data collection.

### EO-79 / QB50 P1 (FUNcube 3)

2U CubeSat with UHF to VHF linear transponder. Downlink 145.946-145.971 MHz Uplink 435.0723 – 435.0473 MHz
No AMSAT telemetry, but primary system beacons on 145.815MHz
Launch: DNEPR from Russia. 19th June 2014
Orbit: Sun synchronous orbit at 600km
Collaboration: Von Karman institute for fluid Dynamics, Innovative Solutions In Space (ISIS bv) Delft Netherlands, AMSAT-NL and AMSAT-UK

QB50 P1 and QB50 P2 were technology demonstrators, or precursors for the QB-50 project. These 2U satellites tested the design of the science instruments for the QB50 mission. In acknowledgement of the use of amateur radio spectrum for the 50 CubeSat mission, the Von Karman Institute agreed to include amateur radio transponders on the 2 precursor satellites. These were to be activated after the primary mission had been completed.

**Current Status**

The EO-79 transponder was made available for amateur radio on a regular basis from 12th November 2016. However, the power consumed by the satellite with its attitude control system and other payloads have left the CubeSat with a potentially negative power budget. If the transponder is switched on, the battery is depleted within 4 or 5 orbits. To combat this power drain issue, engineers at ISIS developed and uploaded new flight code to the satellite. This code detects the transition from eclipse to sunlight and allows transponder activation based on programmable timers.

Transponder timing as implemented on EO-79

T=0 Satellite exits eclipse. Typically over The Southern Ocean. Delay timer starts / battery charges in sunlight.
T=27 mins. Transponder activates. 25 min transponder timer starts. – Satellite is on ascending pass travelling S to N
T = 52 mins. Transponder switches off after 25 mins on time. Satellite is now typically at 55 degrees north.

The satellite then has a period of apx. 10 minutes in sunlight for it to add some charge to the battery. It passes into eclipse with the transponder off but with enough charge in the battery to operate until the next period of illumination.

**EO-88 / Nayif -1**

UHF to VHF linear transponder. + BPSK Telemetry @ 1200bps on 145.939 MHz
Launch: PSLV-C37 from Sriharikota, India. 15th February 2017
Orbit: Sun synchronous orbit at 495km
Collaboration: Mohammed bin Rashid Space Centre (UAE) and American University of Sharjah,
Innovative Solutions In Space (ISIS bv) Delft Netherlands, AMSAT-NL and AMSAT-UK

**Current Status**

Nayif-1 is owned and operated by the United Arab Emirates. This educational satellite carries an upgraded FUNcube transponder which includes a higher stability oscillator design. The satellite has active attitude control and features the usual 'fitter' text messages but with optional Arabic language. Although manual commanding is possible, currently the satellite transponder is operating entirely automatically , using eclipse or sunlight detection. The spacecraft operations can only be described as 'nominal'

**Future Projects**

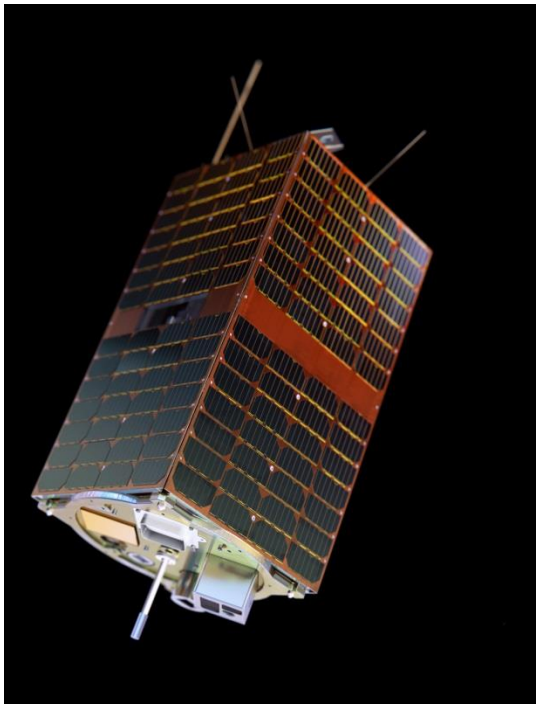
JY1-Sat is a 1U Jordanian educational satellite project. The satellite, an initiative of The Crown Prince Foundation, is currently under construction in Amman Jordan and The Netherlands. The launch is planned for early 2018 from a Falcon 9 rocket.

JY1-Sat is named after the late King Hussain of Jordan who held the callsign JY1. The satellite will carry another FUNcube UHF to VHF linear transponder with enhanced oscillator stability. The satellite also includes active attitude control.

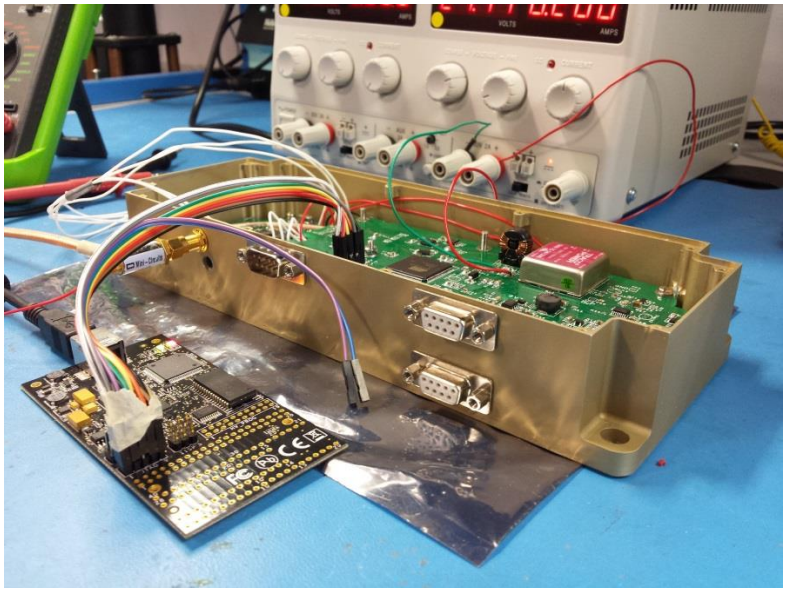
JY1-Sat will transmit stored images reflecting Jordanian culture and its historical heritage. The images will be transmitted in a digital (SSDV) format along with the standard FUNcube BPSK telemetry. Images will be decoded and displayed on the JY1-Sat Dashboard software.

# ESEO - European Student Earth Orbiter.

This 50kg satellite is a 'Hands on Project' from The European Space Agency Education Office.
The project has been running in several incarnations for about 10 years, but the spacecraft is now undergoing final integration. It is a collaboration between ESA and the prime contractor Sitael in Italy. Sitael are providing the structure and core satellite systems while Universities from around Europe, together with AMSAT-UK are providing a number of Science and communication payloads. ESEO is 3 axis stabilised, incorporating attitude control using a combination of magnetorquers, momentum wheels and Nitogen gas thrusters. Communication between the satellite OBDH and the payloads uses a CAN bus, based on the CANopen protocol.



**ESEO. 50kg 40cm x 40cm x 60cm**



**AMSAT Payload (EM) being programmed at Surrey University**

**Objectives of the AMSAT payload**

1) Educational outreach via telemetry on 145.895MHz using 1200bps in a similar way to AO-73
2) An FM transponder with L band uplink on 1263.500MHz and downlink on 145.895MHz - CTCSS 67Hz
3) Provide a back up communication system for transmitting Science data at 4800bps (primary system is 2GHz)
4) To test new technologies e.g 32bit processing and Silicon Labs factory programmed LVPECL oscillators for the L band local oscillators and a CMOS VCXO for the FM transmitter.

The telemetry and transponder downlinks share the same 2m frequency. This design decision was made to preserve available spectrum on the crowded 2m band, but also to reduce the payloads power budget and provide some redundancy in the design by using separate BPSK and FM transmitters. The payload block diagram in the presentation shows how the 2 transmitters are switched between a common low pass filter and 2m antenna using a GaAs switch from Skyworks.
ESEO will be launched into a LEO sun synchronous orbit around 550km with a high orbital inclination.

### Transponder receiver and transmitter characteristics

To access the ESEO transponder, a user will need to transmit a carrier that is on 1263.500 MHz at the satellite. A 67Hz CTCSS tone will normally be required. Doppler shift on L band is anticipated to be plus and minus 27kHz, so finding the correct frequency by hand is going take practice. The receive antenna on the satellite is a right hand circular polarised patch which is printed on a PTFE substrate. With 3 axis stabilisation, the patch will usually be nadir pointing but as the antenna has gain, you may find that ESEO needs to be several degrees above the horizon before you are in the main lobe of the patch antenna.
The L band receiver is a triple conversion superhet design with intermediate frequencies of 45MHz, 10.7MHz and 455kHz. Filter bandwidth of the 455kHz stage in the transponder receiver is 15kHz. Uplink audio deviation around 3kHz should be sufficient, and testing has shown that a signal of -112dBm at the receiver will produce a satisfactory output.
The receiver's 67Hz tone detector requires 500mS of tone before the transmitter is activated. If the 67Hz tone is lost, the transmitter will remain on for a period of 20 seconds. During this time it will retransmit whatever signals are present at the receiver input.
The FM and the telemetry transmitters are both stable over a wide range of temperatures. Thermal vacuum testing covered multiple cycles between +70 degrees and -20 degrees centigrade. The transmitters can both be commanded into a high or low power output, with transponder signals on 2m being comparable with AO-85.
The 2m transmit antenna is constructed around a Teflon coated multistrand wire which is supported inside a 3mm dia carbon fibre tube. Due to restrictions of the launch adaptor the 2m antenna is mounted on the space pointing end of the satellite.

### Telemetry transmission

The normal telemetry mode will be 1200bps BPSK with forward error correction. The format is similar to AO-73 with 4.3 seconds of data followed by the 700ms characteristic 'beep'. Each frame contains 2048 bits of data. This incudes an 8 bit satellite ID and an 8 bit frame type indicator. This is followed by 432 bits of real time telemetry and then 1600 bits of either whole orbit data or a text 'fitter message'. The 2048 bits are expanded to 5200 bits by applying FEC and at a transmission rate of 1200bps take 4.3 seconds to transmit. With 700mS of tones the total frame time is 5 seconds.
However, unlike AO-73, only 4 text message slots of the 24 available have been included allowing more time for sending telemetry. Data is gathered internally within the payload, using I2C data transfer between the sensors and the processor. Telemetry collected from the main satellite systems, including the electrical power system and several 32 bit channels generated from magnetometer data is transferred via CAN.

### Science data transfer

The main functions of the ESEO platform, such as scheduling and telemetry, are controlled via a communication system on 70cms. Science payload data however, is transferred to the ground by a dedicated student communications payload which operates in the commercial space allocation at 2GHz. This system has data transfer rates from 100kbs to 5Mbs and uses a variety of modulation formats. In the event that this system should fail, the science data can be transferred to the ground via the AMSAT payload at 4800bps. When active, the 4800 rate still transmits AMSAT real time data, but the science data replaces the final 1600 bits in each frame. Given the size of the data transfer required (including images) a hole map is used so that data that has not been received in one pass or orbit can be specifically requested for transmission.

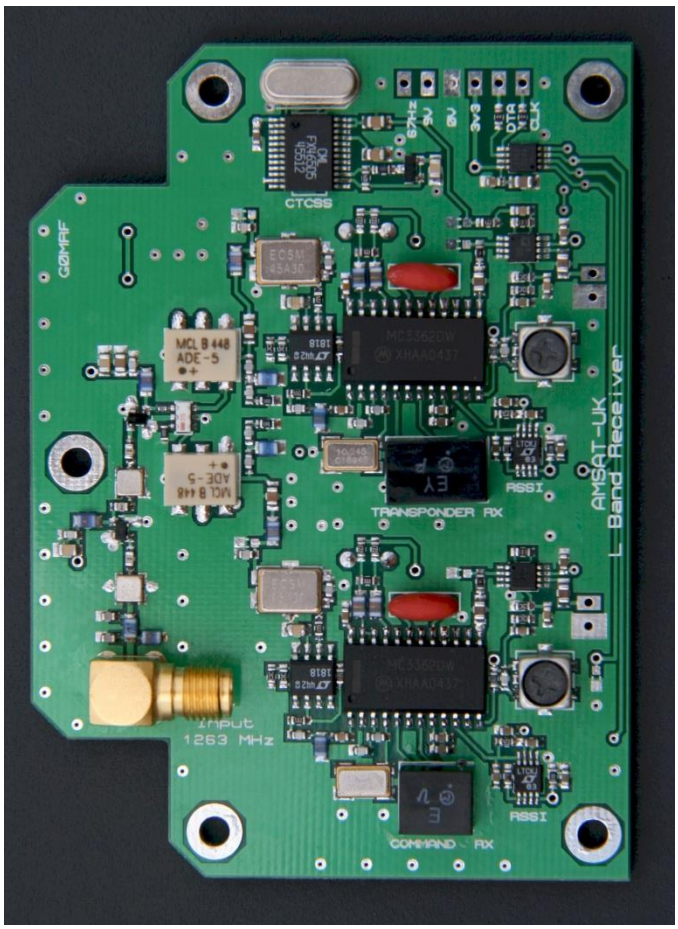
### Operational modes

The AMSAT Payload has 6 main operational modes. Each can be commanded directly from the ground or by the internal satellite scheduler. Additionally some mode transitions will occur automatically i.e. from high power to low power to receive only, if pre-set temperature limits are exceeded. The 6 modes are summarised below:

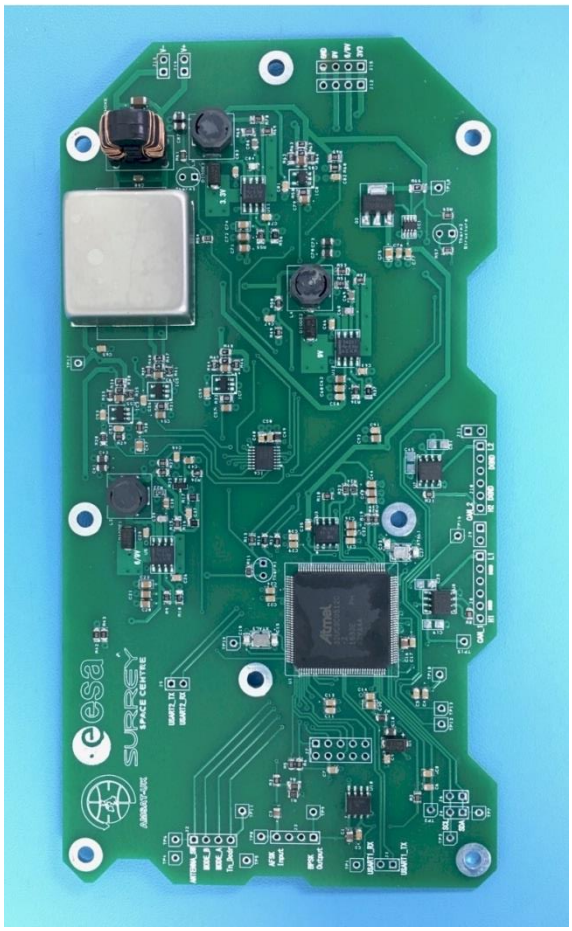
| Payload mode | Function | DC power - Comparison |
|---|---|---|
| Receive only | Transmitters off but with telemetry data collected and stored to memory | 22.0V at 114mA = 2.51W |
| Telemetry mode | BPSK TX on. Transponder off.<br>With telemetry transmitted on 2m | Low power mode 4W<br>High power mode 5.27W |
| Transponder mode | FM Transponder on.<br>Uplink 1263.500 Downlink 145.895 MHz | Low power mode 4.2W<br>High power mode 5.17W |
| Autonomous mode A | Transponder available while satellite is in eclipse. BPSK Telemetry on while in sunlight. | 64 minutes at 4W or 5.27W<br>34 minutes at apx 3W<br>Orbit average (hi pwr) 4.46W |
| Autonomous mode B | BPSK Telemetry transmitted until 67Hz tone received for 0.5sec on L band, then transponder is activated. System reverts to BPSK transmission 20 seconds after loss of CTCSS signal. | Very similar to telemetry only |
| Science mode | High power BPSK at 4800bps<br>Sends real time tlm + data transfer from a selected science payload. Transponder is off. | 5.27W |

RF output for BPSK telemetry is apx 110mW / 600mW

RF Output for the transponder is 400mW / 800mW




ESEO. Command & transponder L band receivers




Flight CCT Board with 32bit Atmel processor

# Introduction to SatNOGS

by Corey Shields KB9JHU

The Satellite Networked Open Ground Stations (SatNOGS) project is aimed at improving the amount of data collected from cubesats and amateur radio satellites through automation and lowering the cost of entry into satellite ground stations. This is accomplished through open source software that fully automates a ground station, matched with a suite of web applications that manage observations jobs and satellite details. To lower the barrier of entry into satellites we also offer plans for a 3D-printed AZ-EL rotator, antennas, and even a radome - though the modularity of our software is designed to work with most combinations of open and commercial gear. All data collected is provided for free under a creative commons license.

## Goal of SatNOGS

Universities and research institutions have relied on Amateur Radio and satellite enthusiasts to help collect and provide data from their satellites. This often requires that the ground station operator be at their computer to set up the observation by:

1. Configuring their antenna rotator software for the pass
2. Tuning their radio or SDR software
3. Piping the audio through a decoder
4. Uploading the resulting frames to the researchers

This process is repeated for each observation made by the ground station operator.

Meanwhile, cubesats are being launched more frequently and in larger numbers. Ten years ago we saw nine cubesat launches in a year. To date, 279 nanosatellites and cubesats have been launched in 2017[1] (when you account for commercial satellites as well). Dispensers like the QuadPack and SHERPA ensure that cubesats are no longer being deployed one at a time, but now dozens at a time. This presents a big problem for researchers who depend on amateur collection of cubesat data - **the number of satellites are quickly outgrowing the volunteer base of ground stations.**

The goal of SatNOGS is to address this problem directly through writing software that fully automates the collection and reporting process. In the end, a ground station operator no longer needs to be at their computer to control the rotator or adjust the frequency for doppler correction. This means that a ground station can be collecting observations 24 hours a day, 7 days a week.

[1] http://www.nanosats.eu/

## Libre Space Foundation

The Libre Space Foundation (https://libre.space) was founded to make use of Open Source Software principles within the space industry. The mission is to promote, advance and develop libre (free and open source) technologies and knowledge for space. To do that we design, develop and deliver space related projects ranging from ground station equipment to global monitoring networks, rocketry, and satellite missions. The two main projects in the LSF to date have been SatNOGS and UPSat. UPSat is a fully open source cubesat launched in May of 2017.

Claiming space “the libre way” means making data freely available to all. As such, all of the data collected from SatNOGS is freely available to researchers and enthusiasts under a Creative Commons license. We also make APIs available to our web services for other applications to hook into and make use of.

## Hardware

At the heart of the SatNOGS client reference platform is a Raspberry Pi 3. This was chosen for its cost and ability to run as a standalone client in the ham shack, or even be portable enough to deploy in a weatherproof container outdoors. While the client should run on any architecture with a Linux distribution with Python 2.7 and Gnuradio 3.7.10, the development focus is on testing validity for the Raspberry Pi.

A core principle of the SatNOGS project is modularity; we want to make this project cheap to operate while still working with existing ground stations. For rotators this means that the client will work with anything supported by hamlib, which includes most commercial rotators on the market today.

No rotator? No problem. We can make use of ground stations that have fixed antennas or we have plans for a homebrew rotator. This rotator is framed with 2020 extrusion and makes use of 3D printed parts to build an inexpensive option to commercial rotators. The electronics of this rotator implement the easycom rotator protocol and can be used by any software that supports easycom. The design also includes an acrylic weatherproof casing. I can personally attest to this rotator surviving all 4 seasons in Indiana. All-in, the cost of the rotator is around $300, assuming you already have access to a 3D printer.

A homebrew rotator is no good without antennas, so we provide designs for those as well. Circularly polarized helical antennas are a good match with the SatNOGS rotator, while Yagis have also been used. These antennas are designed to be lightweight yet functional for low earth orbit satellites, and built to last. As for fixed antennas, we have had good success with ground stations on our network running Turnstile and Lindenblad antennas. Adding a low-noise amplifier helps in any case of working with satellites, of course.

For signal reception we support the use of many Software Defined Radio (SDR) devices, with our reference platform focusing on the rtlsdr. While many others are superior, the low cost of the rtlsdr is hard to beat when it meets all of our needs. That said, any SDR device supported by osmosdr should work with our gnuradio scripts.

## Online Components

The SatNOGS system is a client/network infrastructure. While the hardware designs can be used as a standalone or portable satellite ground station, the true power comes when a ground station is running the SatNOGS client in conjunction with our web services. More on the client in a moment, let's first dive into what these web services are:

**SatNOGS DB (https://db.satnogs.org):** The SatNOGS Database is a crowdsourced source of truth for satellite data. This starts with functional data such as information about the telemetry beacons on board and what their status is. This site also becomes a repository for all telemetry data collected for each satellite. Modularity in our APIs allows us to interoperate with other applications in and outside of SatNOGS. For instance, gpredict can use SatNOGS DB as a source for satellite and frequency data. The DK3WN telemetry forwarding app now sends decoded data frames to the SatNOGS DB. To date, over 2 million frames of satellite data have been collected (with many thanks to PE0SAT, DK3WN, and all other contributors[2]). This site and its data are available for you to use in your daily satellite interactions.

**SatNOGS Network (https://network.satnogs.org):** The SatNOGS Network is where all of the SatNOGS clients are centrally commanded and controlled. From here, network operators can schedule observations for any satellite marked alive and well in the SatNOGS DB. These observations may span hours and cross many ground stations in the network, giving better coverage for a satellite during crucial times. The scheduling system avoids conflicts in scheduled observations for a ground station, while keeping it busy 24/7. Any data collected from an observation by the SatNOGS client is displayed here in the Network as well before it is sent off to the SatNOGS DB. The Network website is open, anybody can go and browse the current and past observations.

## SatNOGS Client

With the web services in place, the Client becomes a powerful force in the SatNOGS ecosystem. Running on a Raspberry Pi 3 with an rtlsdr and a rotator attached via a USB controller interface, the process of the client works as such:

1. The client checks in with the Network regularly to see if there are any jobs scheduled
2. When a scheduled job occurs,
    a. the client then begins rotation control, tracking the satellite

[2] https://db.satnogs.org/stats

    b. the client starts one of many gnuradio scripts depending on the mode requested
    c. the client continuously corrects for doppler shift in the tuned frequency
3. When the job is over, the client stops moving the rotator and stops the remaining scripts
4. Every few minutes, the client looks for collected data to upload to the Network. This includes
    a. A generated waterfall of the signal
    b. Demodulated audio of the observation (not raw IQ)
    c. Any data decoded by the client script

This process is fully autonomous. Once setup, a ground station operator can walk away and the Client will run continuously without need of maintenance or regular re-tuning. The client supports various modes of demodulating and decoding today, including:

- FM Voice
- CW
- USB (BPSK, etc.)
- FM Data modes
- FSK9k6 (full decode)
- APT (full decode to imagery)

## Project Progress and Roadmap

In the past year we have honed in on developing gnuradio scripts for the client which allows us greater flexibility in demodulation and decoding. Our scripts can now automatically decode and upload APT and FSK9k6 data (G3RUH), with CW and AFSK1k2 soon to come.

The next step is then making use of all of the data that we collect. Currently the frames collected in the Database can be downloaded for use in third party dashboards. Work is underway to begin building satellite-specific dashboards that will display data as soon as it is uploaded, along with historical telemetry data.

## How to Get Started

We are always in need of Web developers, Python programmers, gnuradio programmers, satellite data curators, and of course ground station operators. We have documentation on how to get started with any of these tasks at our wiki found at https://wiki.satnogs.org. You can interact with us via our forums at https://community.libre.space or chat with us in #satnogs on IRC or Matrix at freenode server.

DEPARTMENT OF SCIENCE AND TECHNOLOGY (DOST)
AND
HOLY ANGEL UNIVERSITY (HAU)

# AMSAT-PHL

## Amateur and Weather Satellite Ground Station Development Activities at Holy Angel University (Region 3, Philippines)

**L. M. Almazan (WA6LOS), A. I. Parungao (DV3UZT), J. P. de Guzman (DW3CQK), L. P. Paz (DW3CQF) , R. P. Gopez (DW3CQC), K. S. Gueco (DW3CQI), R. J. Sarmiento (DW3CQB), R. M. Pascual (DW3CQJ), J. A. Arciga, Cali Smile, J. Y. Castaneda, H. P. Yalung**
**[*October 27-29, 2017*]**

### 1.0 Abstract

The Philippines, being a disaster-prone nation, relies on data from satellites for multi- remote-sensing applications. In response, the Department of Science and Technology (DOST) supported the development of research based satellites and ground stations all over the country. One of the paths is the employment of a balik-scientist (returning-scientist) from the United States and form a cadre of students, instructors and engineers at the University of the Philippines (UP) Electrical and Electronics Engineering Institute (EEEI) as part of the university's Microsat development program. UP EEEI presented the results and paper of the development at the AMSAT Symposium in 2015. The success of the development has manifested itself to a modification of the program. The huge capital outlay of the original program development and equipment for a local university would be immense. Cost of equipment alone almost doubles due to the Philippine government laws and regulations on importation! This paper describes the accomplishments of the Holy Angel University (HAU) team, located in Central Luzon of the Philippines, in the development of its own inexpensive amateur and weather satellite ground station and the roadmap for some new development in 2018.

## 1.2 Background

The Philippines' geographic location in the Pacific has made it one of the most disaster-prone countries in the world. The country receives more than its fair share of disasters. Typhoons and earthquakes are a common occurrence. Filipino scientists spend great amount of money buying images from commercial satellites for better urban planning, flood control, resource mapping and weather forecasting. These images are not only expensive, but are also limited in terms of quality and quantity. Moreover, the overhead imageries are not in real-time so they are almost unusable for time-critical applications such as disaster response. In response, the DOST supported the Development of Philippine Scientific Earth Observation Microsatellite (PHL-MICROSAT) program with the aim to build, test and launch the Philippines' first earth observation satellite. It addresses not only the need for relevant and timely satellite data for remote sensing applications, but also the need to develop capacity through intensive research and study leading to the actual development of important satellite technologies. PHL-MICROSAT-1 (Diwata-1) was the result and is a joint effort between DOST, the University of the Philippines, Hokkaido University and Tohoku University in Japan and was launched in the summer of 2016 by the International Space Station as a Japan Space Agency payload.

Part of the continuing effort is to involve other universities most especially in an inexpensive ground station development. The Holy Angel University initiative is the first series of ground station development that will be funded by DOST as part of Satellite Ground Station Initiative in 2018.

The Philippine archipelago consists of three main islands: Luzon, Visayas and Mindanao. The DOST initiative, together with UP EEEI Microsatellite Development Center has selected Holy Angel University as the main ground station in Luzon and will house a ground station dedicated to tracking Weather satellites, AMSAT and the Diwata series of LEO satellites. It will also house a network server as repository of received data. Our initial three-month effort at HAU has proven that an inexpensive ground station can be developed by students with the help of faculty members and DOST balik-scientist.

## 2.0 AMSAT/WXSAT Ground Station

The development of the weather satellite and AMSAT ground station provided students, faculties and engineers at Holy Angel University an opportunity to learn about existing weather and amateur satellites; learn to track them and make QSOs. Students were tasked development projects such as, front end filters, Arduino-based antenna switch, automated antenna rotator control, antenna design and parameter testing, network server and the operation of software defined radios (SDR). The team realized the importance of their work on both weather satellite WXSATs and low earth orbit AMSATs which open an avenue for learning research and development, testing components and software algorithms intended for a bigger, more expensive multi-systems satellite ground station.

The development of the HAU AMSAT/WXSAT Ground Station was made possible through the initiative of Leo Almazan (WA6LOS/DU3ZX), a balik-scientist (returning-scientist) from the US Department of Defense (DoD) Space and Naval Warfare Center, together with DOST. A team of ten Electronics Engineering under graduate students and two faculty members formed the Working Integrated Product Team (WIPT). These ten students were divided into five pairs and were assigned different projects. A Program Development Plan and Schedule (PDPS) were developed and subsequent meetings and design review involving the faculty and the team were held in accordance with the PDPS. The following projects assigned were:

- VHF/UHF Cross Yagi Antenna Design
- Bandpass and Notch Filter Design
- Arduino-based Antenna Switch with Graphical User Interface (GUI)
- Adruino-based Azimuth-Elevation Rotator Interface
- Software Defined Radio (SDR)
- Network Server

Figure 1, shows the block diagram of the HAU's AMSAT/WXSAT Ground Station System. The system is intended to receive: telemetry, track and receive satellite data using front-end filters for the SDR. A baseline spectrum analysis inside and outside the WXSAT and AMSAT Lab showed two radio frequency interference sources that de-sensed the very high gain SDR front-end. One was the VHF radios from the school security force and the school FM radio station which was only less than 10 meters from the lab. Another surprised was the interference from a digital broadband cell tower site next to the lab! It was interfering with our ham club HF operation! Further studies with the cell provider and the National Telecommunication Commission, the Philippine equivalent of the FCC, are being planned and a mitigation plan is in the draft stage. What is stunning is that the interference is only in the HF band and not in either VHF or UHF. Figure 2, shows the spectral and waterfall signal analysis of the interference. Note the 50 KHz spacing, what is interesting is the waterfall signals show data. We asked around the university if anyone is performing any experiment in the HF range and there was none. These signals are transmitted 24/7!!!

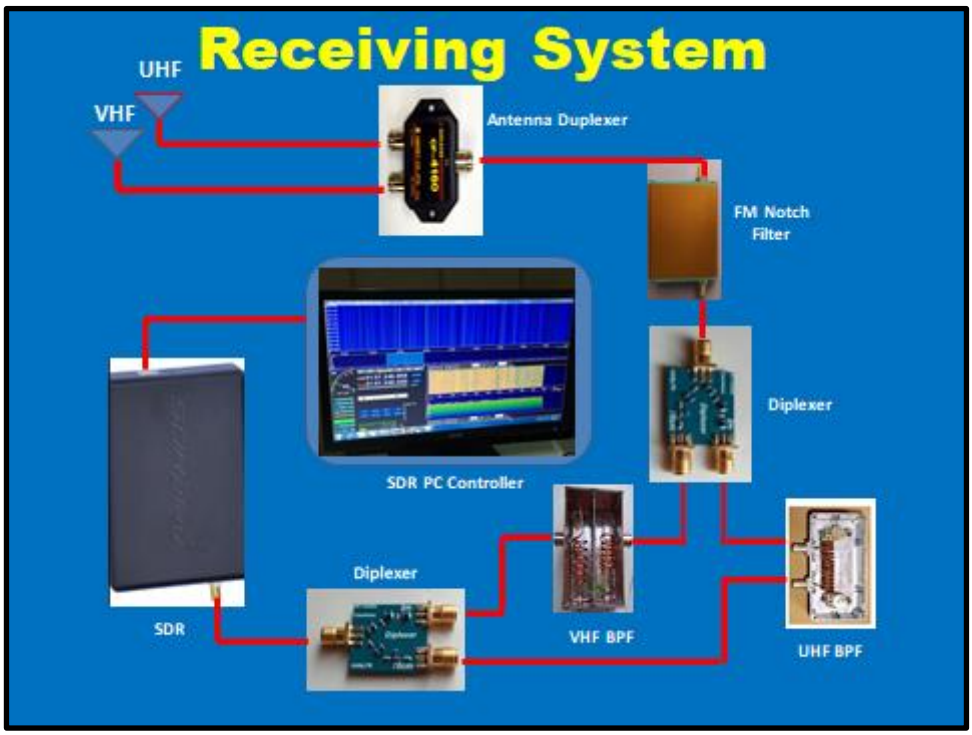



**Figure 1. WXSAT/AMSAT Receiver System**

**Figure 2. RFI on 40 Meters**

## 2.1 Cross Yagi Antenna Design

The antenna used for the AMSAT/Diwata satellite receive system is a cross-Yagi which is made up of two Yagi antennas. The cross-Yagi in this project is a combination of a VHF and UHF antennas which are 90 degrees off from each other. The antenna operates on both 145 MHz (VHF) and 430 MHz (UHF). The antenna was designed using the MMANA-GAL antenna modelling software and the simulated design showed the VHF antenna with a gain of approximately 6 dB and the UHF antenna with approximately 9 dB gain. The antenna simulation segment design and parameters are shown in Figures 3 and 4 respectively.

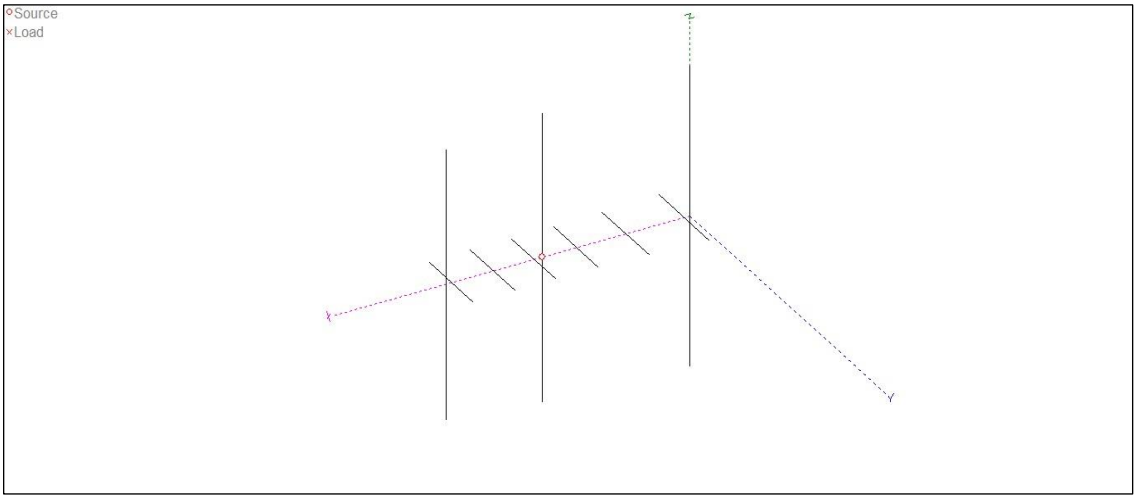



**Figure 3. Cross Yagi Segments**

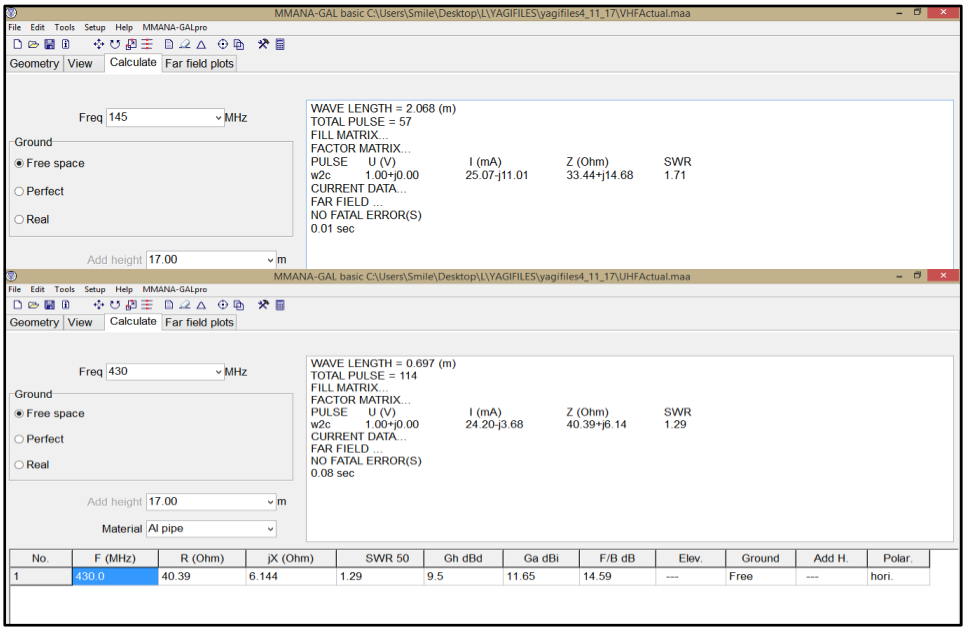



**Figure 4. Antenna Parameters**

The actual antenna was made of scrap aluminum elements and antenna element plastic mounts from discarded TV antennas and another scrap square aluminum tubing for the boom. The antenna was matched with the 50-ohm coaxial line, and used open-stub technique for impedance matching. Antenna testing was conducted outside the lab using a commercial antenna analyzer. SWR, impedance and resonant frequency were measured and the antenna worked with an SWR of less than 2:1 across the desired bandwidth after some trimming of the stub.

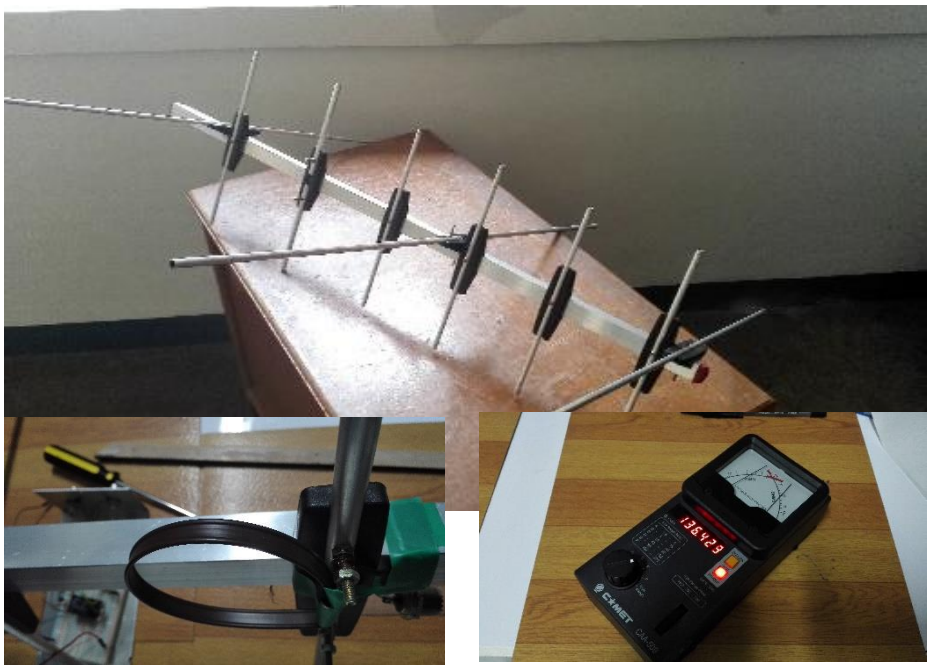


**Figure 5. VHF/UHF Cross Yagi Antenna**

**2.2 Bandpass Filter Design:** The SDR is capable of receiving from 10 KHz to 2.0 GHz and has plenty of front-end gain, but for receiving a small bandwidth in both the VHF and UHF band, front-end filtering was required. Three separate filters were needed, one for the school FM station transmitting on 99.85 MHz and the other two were on VHF and UHF.

The student researchers used an existing circuit design found on many publications and on the World Wide Web. It is a helical bandpass filter (BPF) using a 12-gauge copper coil and an FR-4 PCB enclosure. The circuit was first simulated using Elsie RF Filter CAD software shown in figure 5 and similar CAD software, Simetrix shown in figure 6.

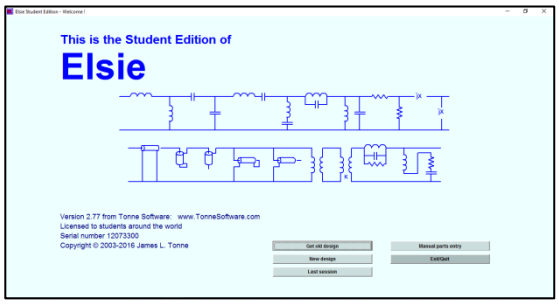



**Figure 5. Elsie Student Version**

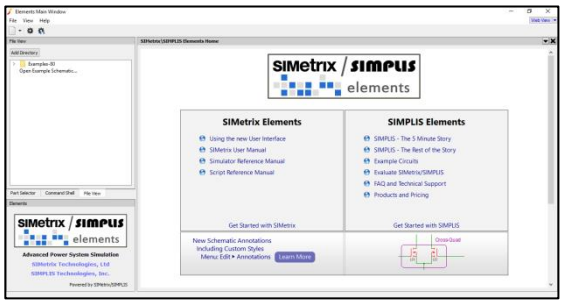



**Figure 6. Simetrix**

The schematic diagram of the VHF BPF is shown in Figure 7. It is a second order filter and has a dimension of 120 x 90 x 50mm.

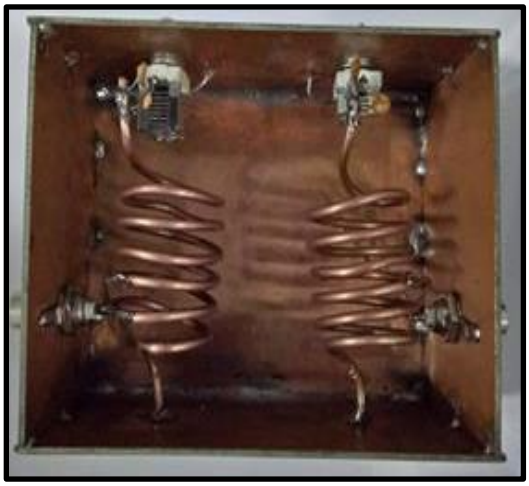


**Figure 7. BPF prototype**

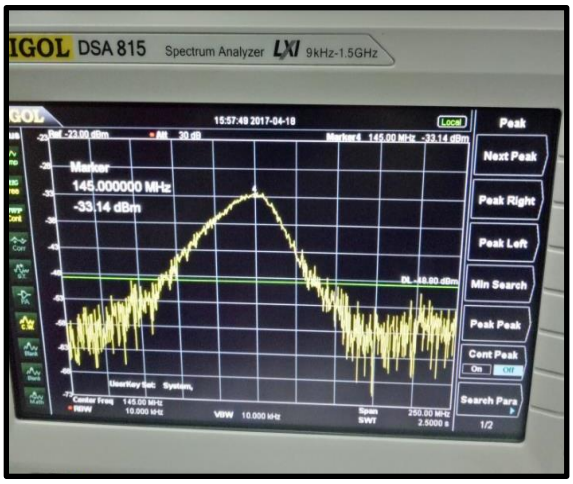



**Figure 8. Band pass Filter Output**

Figure 8, shows the output of the filter that was tested using a scalar network analyzer. It has a center frequency of 145MHz. The researchers learned many aspects of VHF/UHF most specifically the parasitic effects of passive components' leads on those higher frequencies.

**2.3 Notch Filter Design**

A notch filter, also referred to as a band stop or band reject filter, was built to attenuate the university students' FM station transmission. The notch filter shown in figure 9 was designed to reject frequencies between 88 MHz- 108MHz with an attenuation of around 50dB to 60dB. The

signals transmitted by the FM stations greatly affected the SDR unit and the notch filter was used to prevent the overloading of the front-end of the SDR unit.

A Chebyshev type 3-pole filter was designed and built. Its center frequency is 99.85 MHz with an attenuation of -54.13dB. For the schematic drawing, Eagle software was used and shown in Figure 10. The design was simulated using Elsie to verify the characteristic and behavior of the filter as shown in Figure 11.

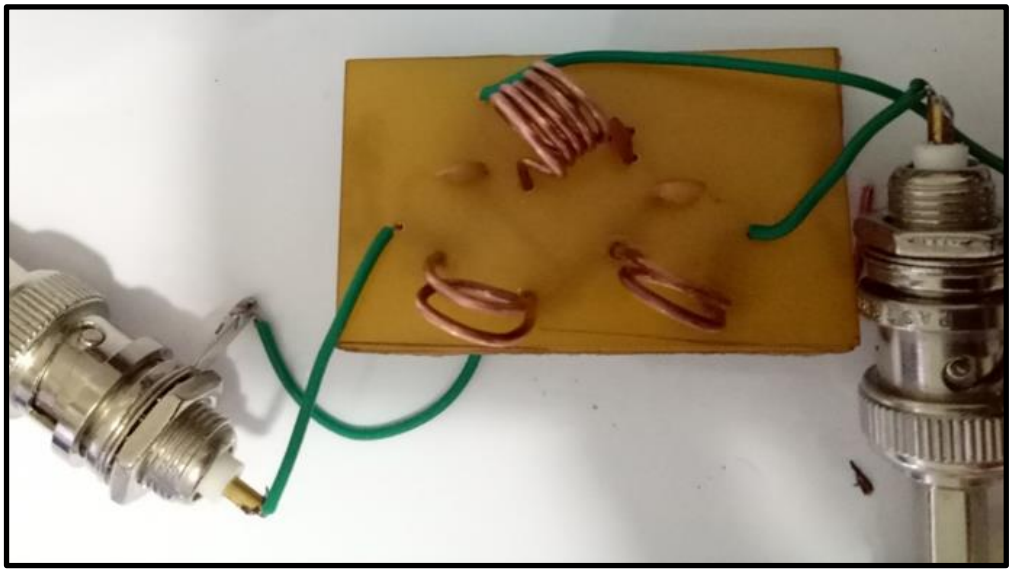


**Figure 9. 88MHz- 108MHz Notch Filter**

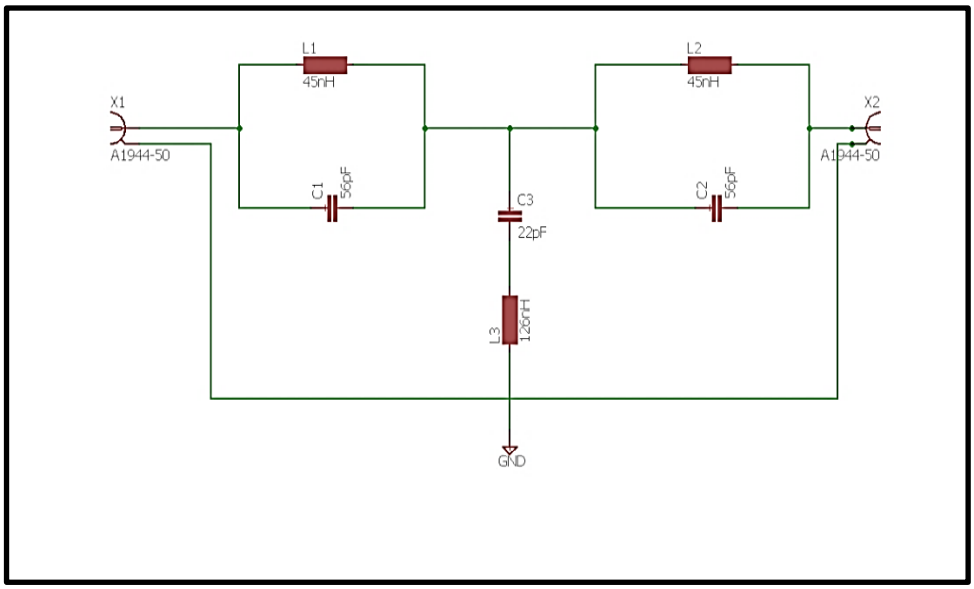



**Figure 10. Circuit Diagram using the Eagle CAD Software**

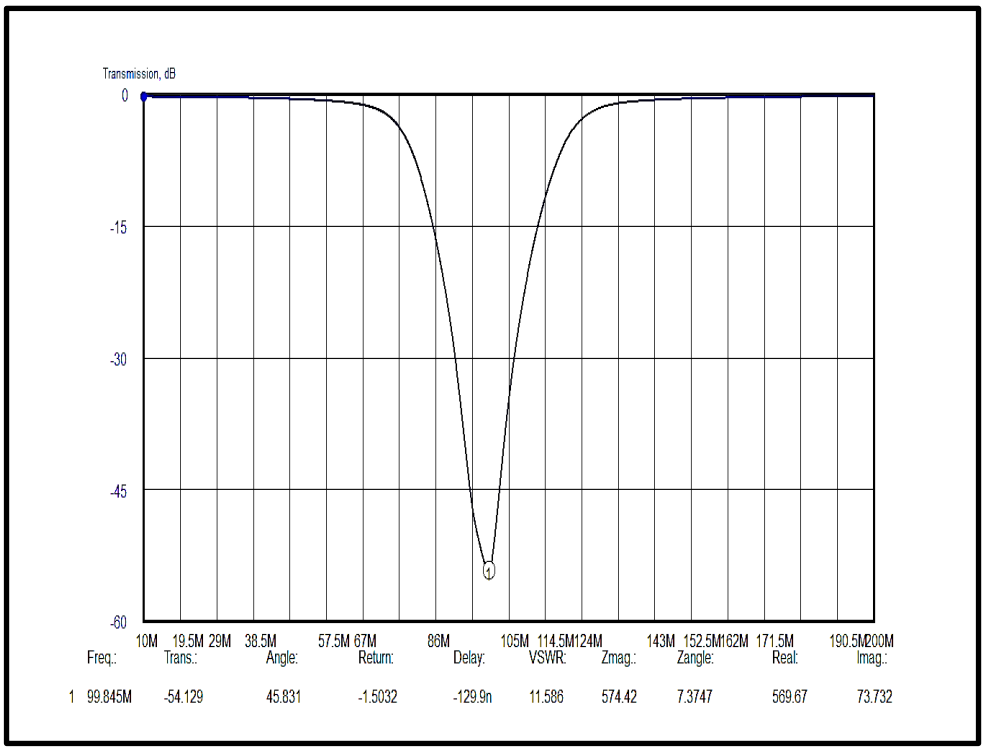



**Figure 11. Notch Filter Output Waveform Simulation (Elsie)**

A 3-D FreeCAD drawing of the filter inside its enclosure is also shown in Figure 12. The notch filter and all the other filter designs were done using freeware CAD programs on the Internet.

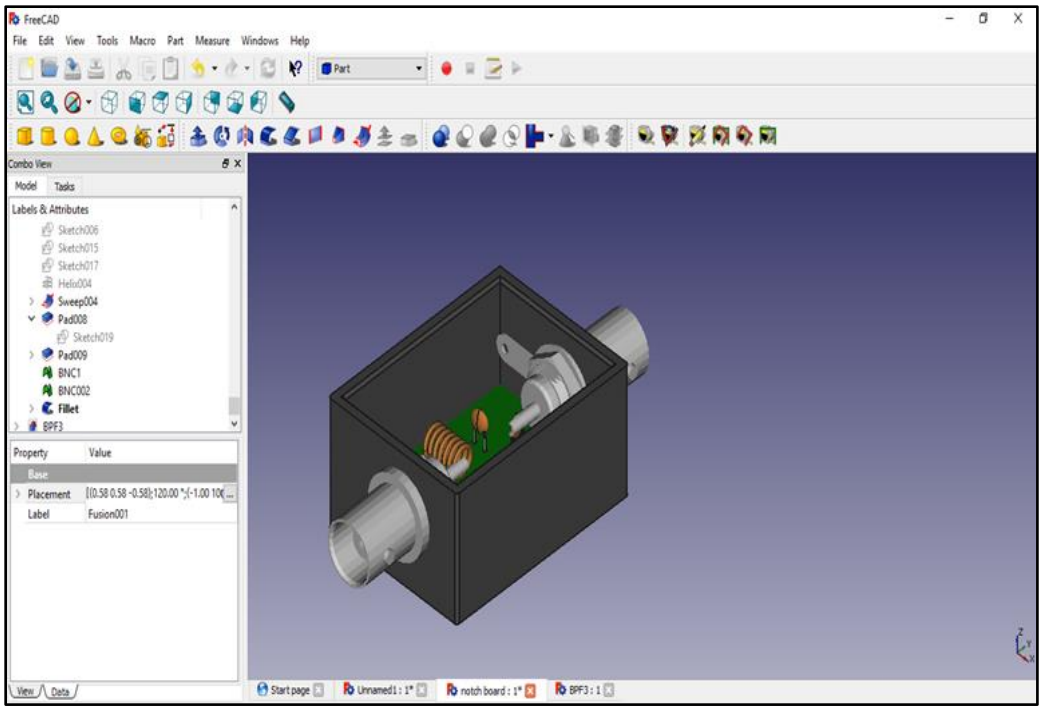



**Figure 12. Three-dimensional Design of the Notch Filter Using FreeCad**

The design was tested using a scalar network analyzer as shown in Figure 13. This was the first time that the students have used a network analyzer and after a brief moment on YouTube, they have become experts in the use of the test equipment!

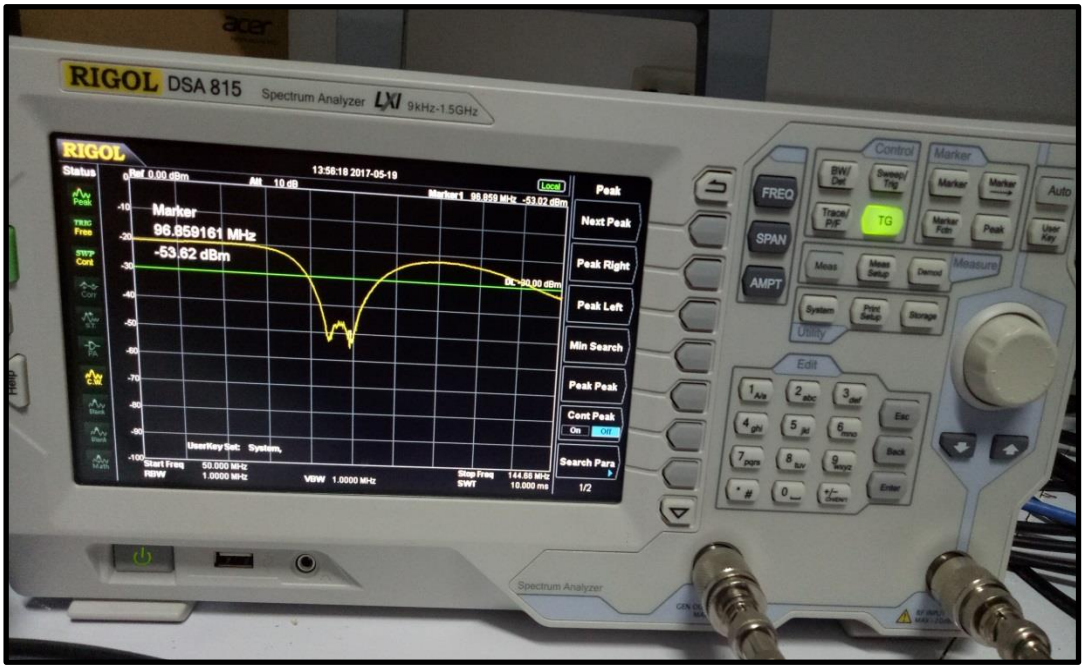



**Figure 13. Notch Filter Testing**

## 2.4 Arduino-based Antenna Switch with GUI

Our ground station also houses our Ham Radio Club station (DX3HAU) and has an HF and a VHF transceivers. Installed on the 4th story roof top are a 40M dipole antenna and a (20/15/10M) tri-band yagi antenna that will be procured in 2018 as part of the DOST initiative. A two-port antenna switch was designed and built to switch either antenna depending on band operation. Figure 15, shows the schematic diagram of the unit. An Adruino Uno development board was used as the microcontroller to automatically switch the desired antenna.

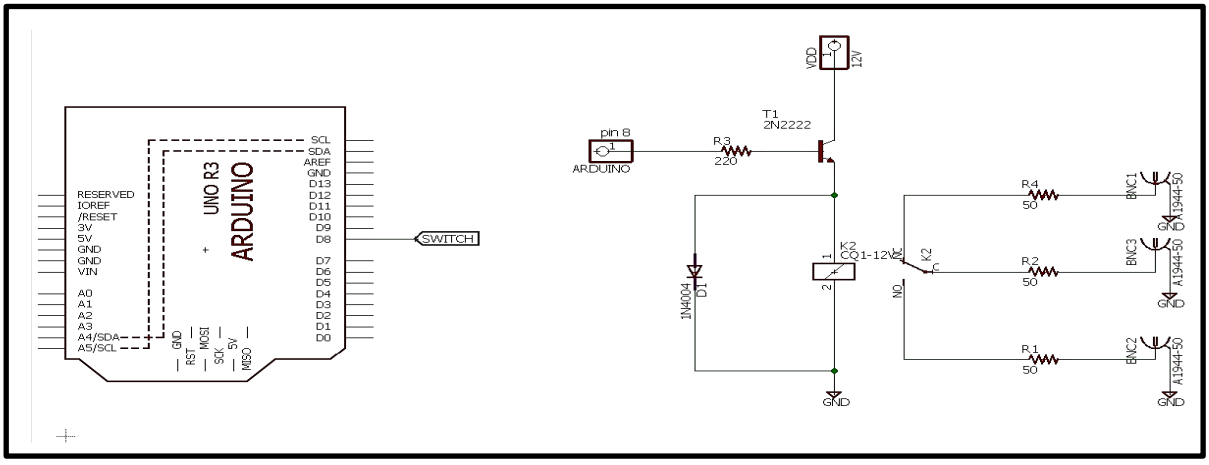



**Figure 15. 2 x 1 Antenna Switch Schematic**

A miniature general purpose single pole double throw (SPDT) relay which was chosen and has a switching voltage of 6VDC and switching capacity of 7A, robust enough so one can transmit more than 100 watts output into the antenna switch. Using a 50 ohm dummy load to simulate an antenna, an SWR of 2.1 was measured at 29.857 MHz and a minimum of 1.3 to 1.5 SWR for frequencies of 1.503 to 21.150 MHz were also measured and shown on Figure 16.

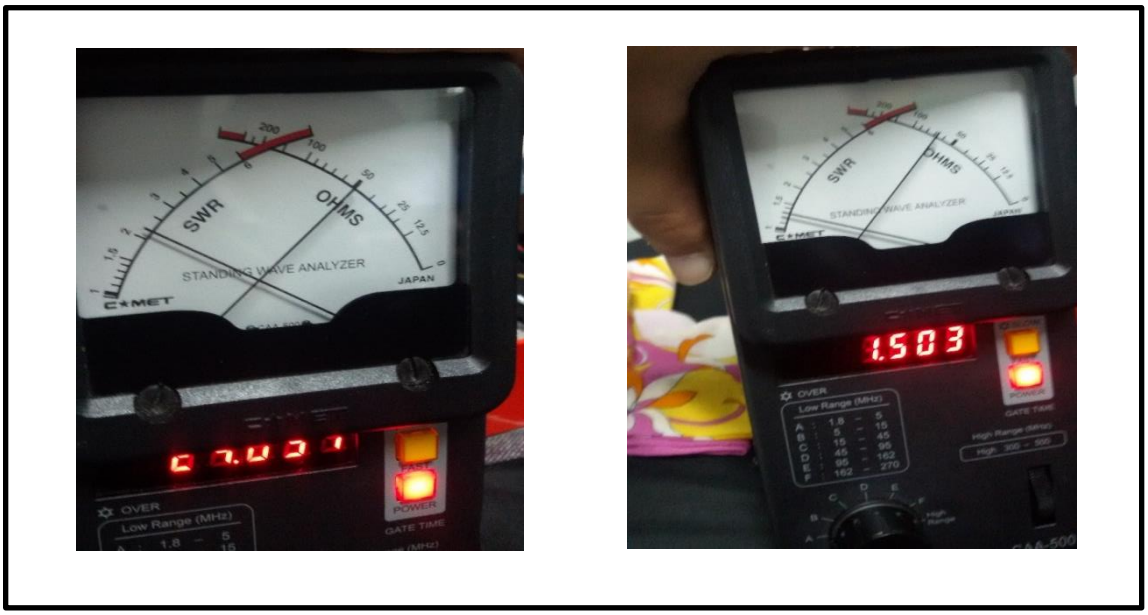



**Figure 16. SWR Measurements**

Additionally, Visual Studio 2017 shown in figure 17 was used in order to create the graphical user interface (GUI) for the antenna switch. Visual Studio is capable of using multiple programming languages, such as C++ language that the Ardruino Uno uses. The basic windows interface was chosen in order to produce a simple GUI for antenna selection.

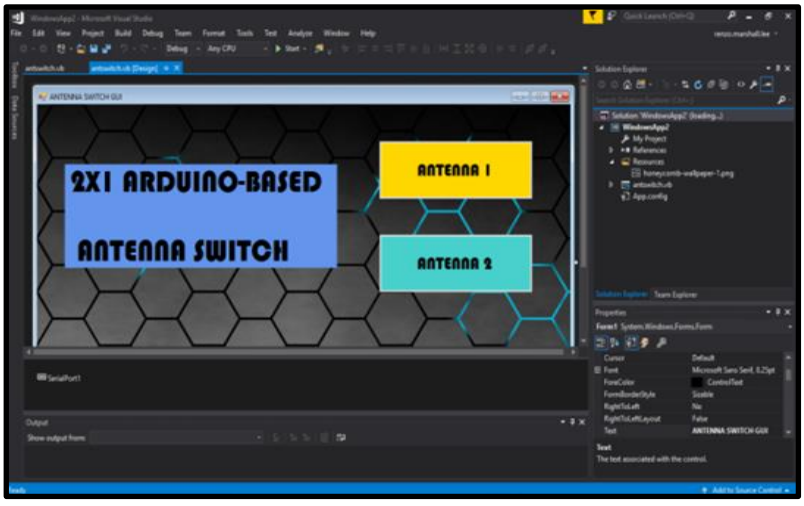



**Figure 17. GUI created using Visual Studio 2017**

## 2.5 Satellite Ground Station Receive System

Our Satellite ground station uses a satellite receive and tracking systems to track and receive weather satellite imageries and data from LEO AMSAT satellites respectively. The system consists of the following:

- Radio Receive Unit
- Radio Interface Software
- Satellite Tracking Software
- Weather Satellite Decoding Software
- Antenna AZ/EL Rotator Unit (under development)
- Right Hand Circular Polarized Antenna (under development)

The radio equipment used for this setup is the SDRPlay RSP1 shown in Figure 18. This radio is for receive only. This SDR is wideband which covers frequencies from 10 KHz up to 2GHz. It is supported by Windows, Linux, Mac, Raspberry Pi 2/3 and Android. It uses 12-bit ADC silicon technology and uses a Type-B USB socket to communicate with a computer and powered through the USB socket. It uses a single SMA connector to connect to an antenna, meaning it has very broadband front-end filtering

**Figure 18. SDRPlay RSP1**

In order to use the SDR, an interface is needed. For this purpose, High Definition Software Defined Radio (HDSDR) was chosen. HDSDR is a freeware SDR program for Windows 2000/XP/Vista/7/8/8.1/10 readily available in the Internet. It is commonly used for general radio listening, ham radio/SWL, Radio Astronomy, NDB-hunting and Spectrum analysis. Figure 19, shows the HDSDR used to control and receive NOAA APT signals at 137MHZ.

The automated frequency control of the HDSDR is a great feature that automatically tunes the SDR frequency based on the Doppler shift frequency of the satellite signal as it passes from horizon to horizon.

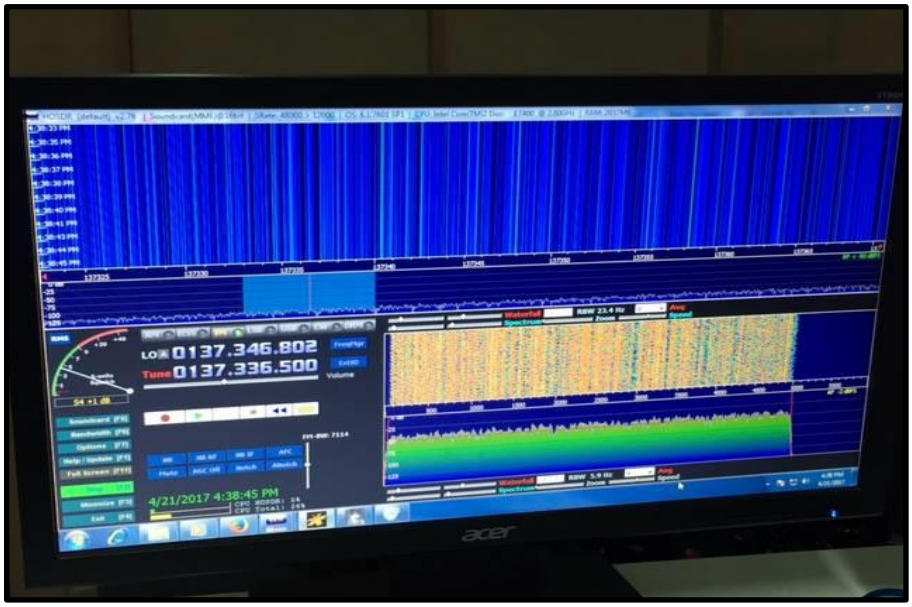



**Figure 19. HDSDR Interface**

Satellite tracking uses the satellite's Keplerian data. To aid in this task, the Orbitron software, as shown in Figure 20, was chosen as the tracking software since it supports simultaneous multiple satellite tracking and has an auto-update option for Two-Line-Elements (TLE) data. It also has a prediction software for the passes and flares of a satellite, an automated antenna rotator control software using WispDD and a simulation feature for predicting satellite orbital path.

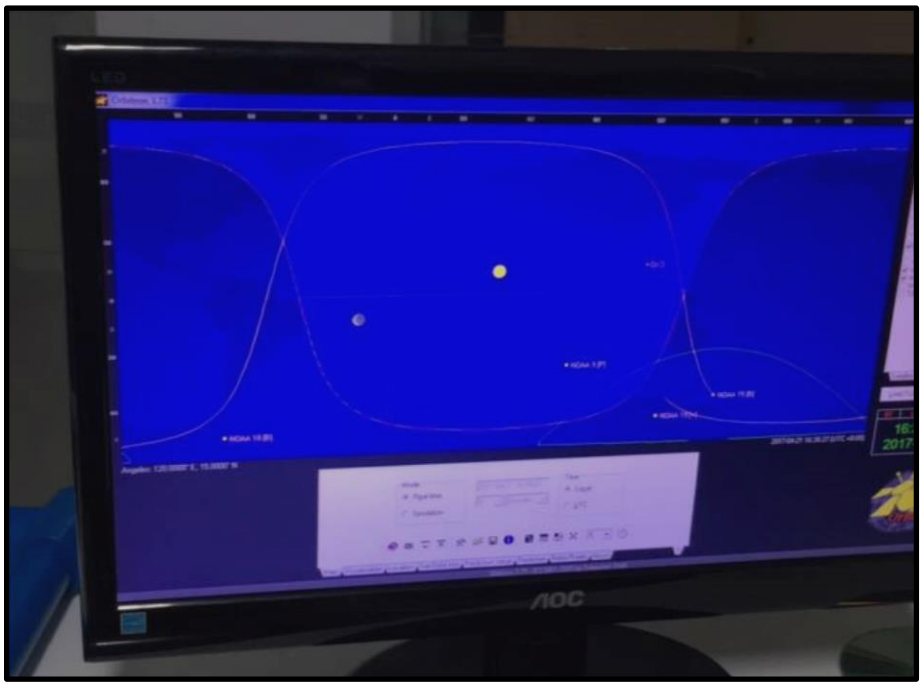



**Figure 20. Orbitron being used to track NOAA satellites**

To decode the received signals, WxtoImg was used as shown in Figure 21. WxtoImg is a fully automated Automatic Picture Transmission (APT) and WEFAX weather satellite decoder. It has multiplatform support including Windows, Linux and Mac OS X. The software is capable of recording, decoding, editing, and viewing weather satellite imageries. It also supports real-time decoding, map overlays, advanced color enhancements and more.

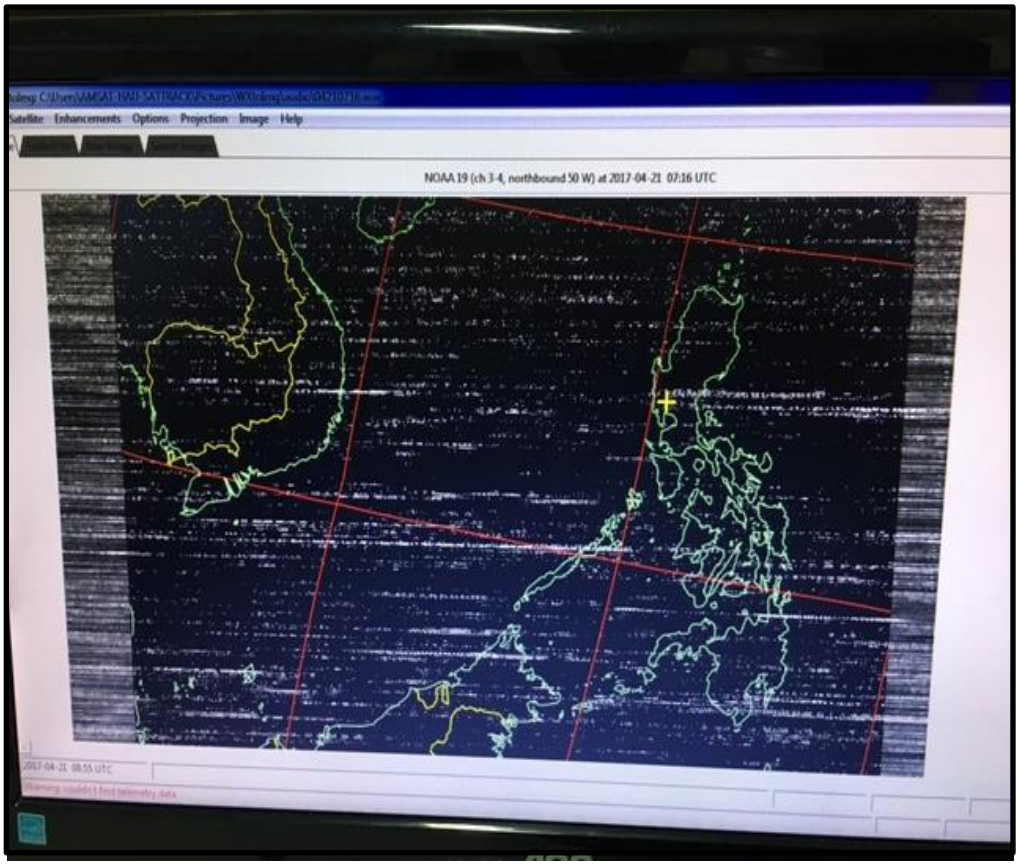



**Figure 21. WxtoImg decoding a NOAA 19 APT**

The first antenna used was a horizontal V-dipole for 137MHz as shown in left side of figure 22. Each leg of the antenna including the connecting wires to the coaxial cable was 53.5cm. The legs were separated by an angle of $120^{o}$. The drawback of this design is that the radiation pattern not omnidirectional. This means that accurate tracking of the satellite azimuth position is a must.

The second antenna shown in the right side of Figure 22 is a quadrifilar helix (QFH) antenna; a better antenna since it is circular polarized and has a wider radiation pattern pointing upwards. Figure 23, shows the photo of the HVD Antenna that was initially tested inside the lab. It was installed on a tripod and was later tested outside and a drop of SWR was noted!

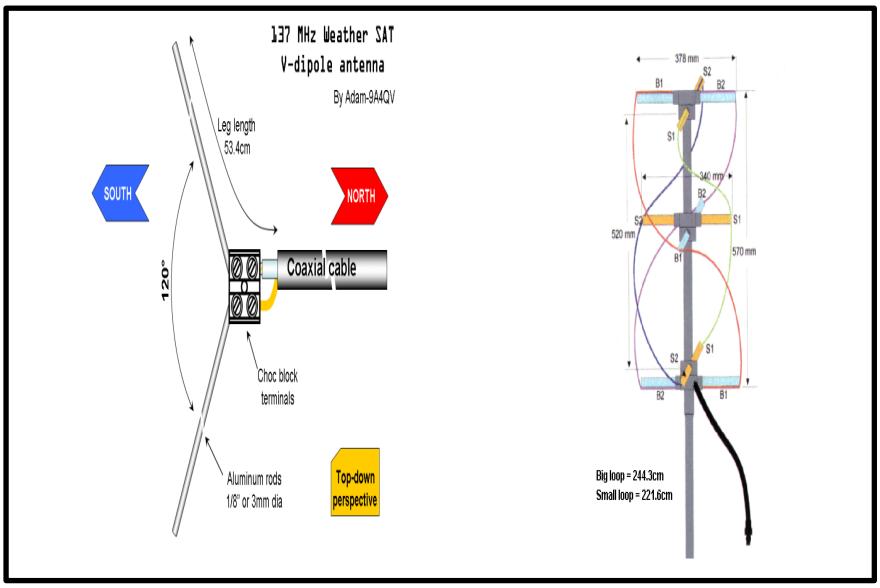



**Figure 22. HVD and Proposed QFH Antenna Design**

**Figure 23. HVD Antenna**

## 3.0 Initial Setup

This is what our AMSAT Lab looks like during our 3-months of development shown in Figure 24. It was built specifically as satellite ground station and to house the university's amateur radio station. It has a large custom built console cabinet that is geared to house 19-inch rack units. It has plenty of lab benches and a good size table for meetings and studies. The lab also houses the Network server that will be tied to the selected ground stations in the Visayas and Mindanao area and will be opened to the AMSAT community in the near future. A back-up server is housed at the UP EEEI Microsat Development Center.

The ground station consists of three flat-screen displays. The one on the left is the satellite tracking display using Nova, one of three satellite tracking software on hand. The middle display is the SDR PC Interface and control display. The one on the right is the APT decoded imagery display.

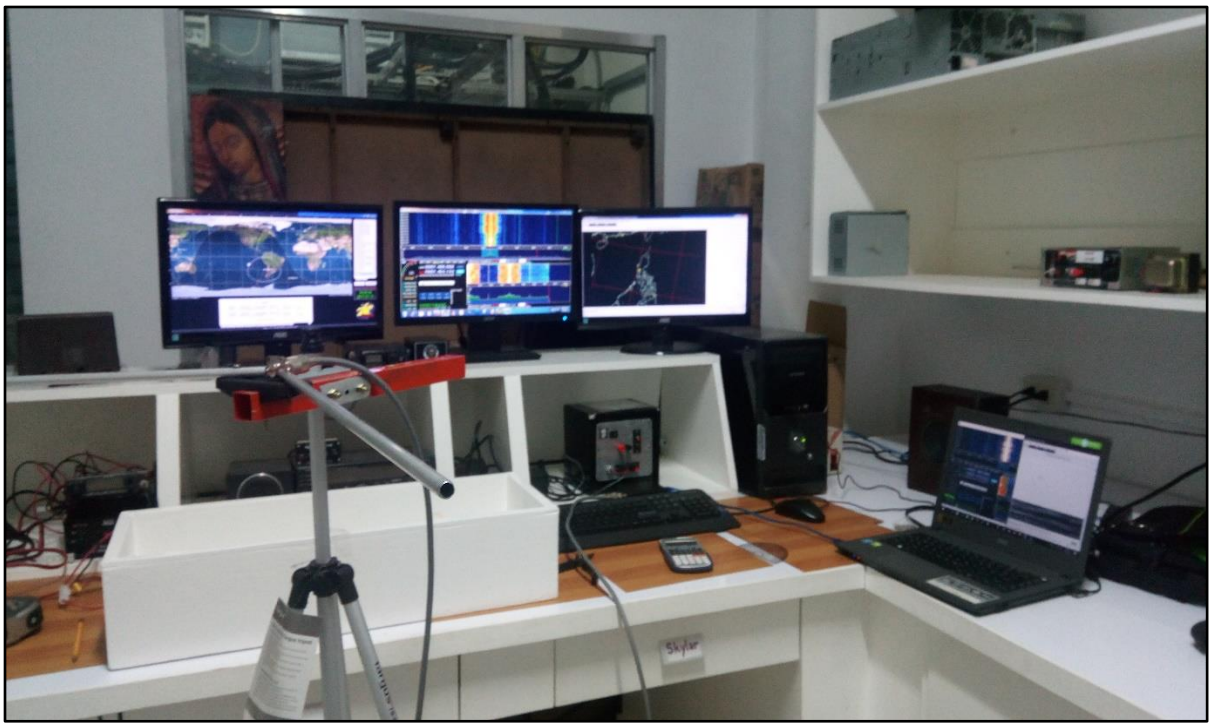


**Figure 24. Satellite Tracking System Setup**

## 4.0 Way Forward

Our next step is to make improvements on the ground station. APT signals are still not 100% received thus resulting in incomplete decoding of images as shown on the left of Figure 25. The student researcher team aims to receive and decode an image like the one on the right of Figure 25. A student development proposal to continue the research and development of the ground station will be submitted to DOST for evaluation before the end of this year.

The other project that needs to be finished is the azimuth/elevation rotator control for our AMSAT satellite station. Additional funding is being sought to purchase a Yaesu G-5500 and continue the development of the PC controlled antenna AZ/EL control for the Yaesu G-5500.

We look forward to 2018 where DOST will fund the rest of the needed development and our proposal of developing a transportable AMSAT and Weather Satellite system that we can transport and deploy in the field when there is an impending disaster similar to the super Typhoon Haiyan in 2013. For 48 hours the town of Tacloban in the province of Leyte in the Visayas had no communication and weather updates from the country's main weather center PAGASA which led to the death of about 6,400 persons and more than 100,000 left homeless because the warning of a storm surge was never disseminated to the local government units in Tacloban due to communication failures. A 100W amateur radio HF transceiver, VHF transceiver and portable antennas similarly will be deployed as part of the deployable disaster kit. It will be used to communicate with the Philippines Inter Island communication network as part of the NTC and Philippine Amateur Radio Association's Ham Emergency Radio Operation (HERO) network.

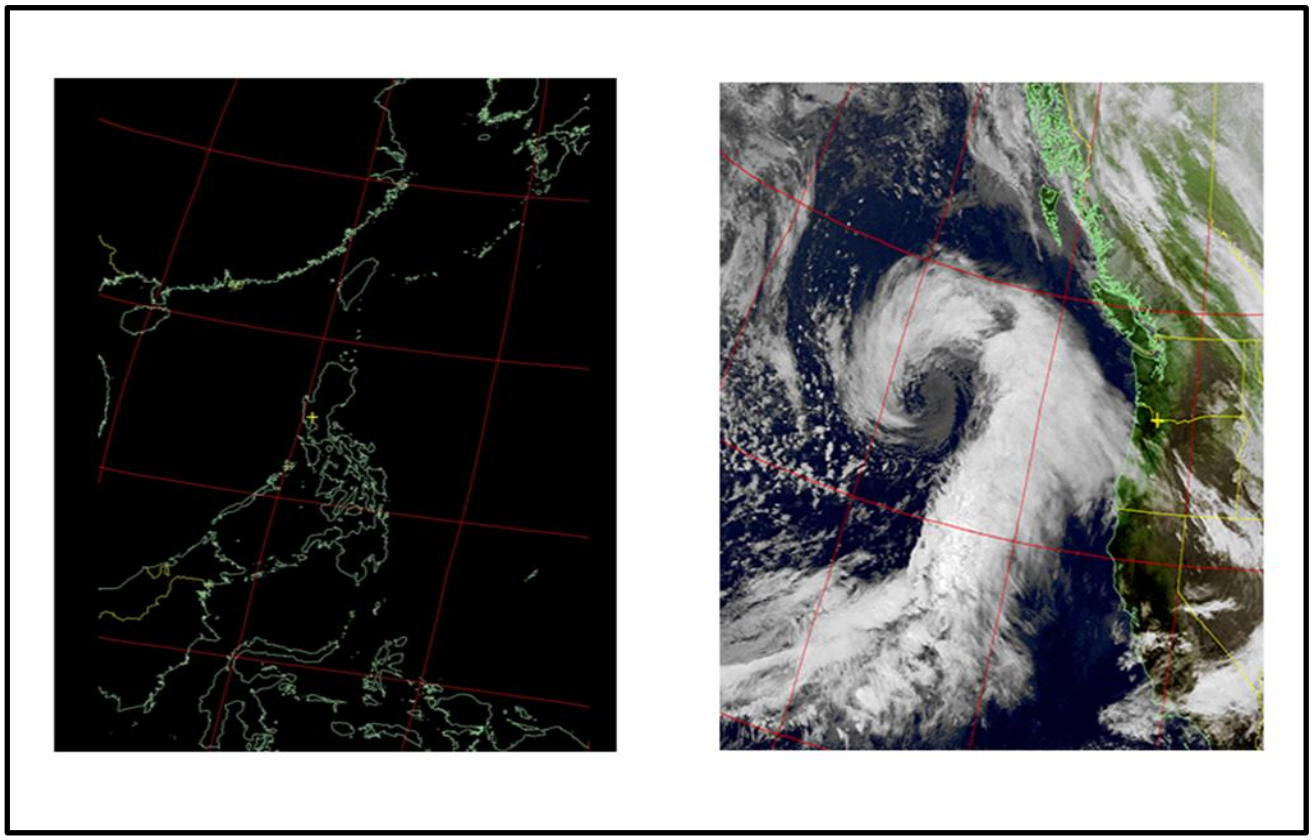


**Figure 25. APT Decoded Images**

**REFERENCES AND ACKNOWLEDGEMENTS**


[1] A. Gonzales, J.M. Oliveros, J.L. Labrador and J.P. Espiritu, PHL-MICROSAT Team in Tohoku University, Japan

[2] Department of Science and Technology (DOST) Balik-Scientist Program. Available: http://bsp.dost.gov.ph

[3] ARRL Satellite Handbook, Ford, First Edition

[4] Radio Society of Great Britain, VHF/UHF Handbook, 1985

[5] M. Spencer, WA8SME, *"Satellite CAT Interface for Working Analog Birds"*, ARRL QST

[6] Indigenous Satellite Development for the Republic of the Philippines Concept of Operations L. M. Almazan, January 2014

[7] WXTOIMG, http://www.wxtoimg.com

[8] SDR-Play-1, http://www.sdrplay.com

[9] NOAA Weather Satellite Tutorial. https://www.rtl-sdr.com/rtl-sdr-tutorial-receiving-noaa-weather-satellite-images

[10] Tracking NOAA Satellites. Available: https://www.n2yo.com/satellites/?c=4

[11] Adruino SimpleSat Rotor Control. Available: https://amsat-uk.org/info/simplesat-rotor-controller/

[12] Receiving weather satellite images. Available: http://www.g4ilo.com/wxsats.html

# Albuquerque Public Schools
# STEM Trajectory Initiative

## CubeSAT Project

**Presented By** Alma Ripley, Assistant Principal
Albuquerque Public Schools, Albuquerque, NM
Ripley_A@APS.edu

**What is the STEM Trajectory Initiative?**

STEM Trajectory Initiative (STI) is:

- Four years of changing students' minds about their ability to be scientists, technicians, engineers, and mathematicians.
- Four years of influencing a shift in teacher's mind-set as they demonstrate excellence, commitment, and go over-and-beyond their "job description", day-after-day, to deliver project-based learning opportunities through STI.
- Teaching students the bridges between language, writing and STEM.
- Showing students that they can reach the International Space Station, and explore the outer edges of the atmosphere.
- Involving hundreds of students from early elementary school through high school graduation, and beyond, those students returning to APS in the summer to give back to APS as they continue participating as presenters during the Dual Credits Electronics Courses and remain committed in the multi-year STI CubeSAT Project.
- Aligned with Early Learning, Developing the Whole Child and College & Career Readiness.
- Teaching collaboration, writing, and critical thinking.
- Completing over 100 successful STI multi-school projects.
- Partnering students with their communities with a specific focus.
- Showing students that their world is larger than they ever knew.

## Background

At the beginning of SY2013-14, the state assessment; Standards Based Assessment (SBA), was on its way out and PARCC was the new "buzzword" around APS. "No Child Left Behind" was being left behind, and "Every Student Succeeds Act (ESSA) was signed into law in December 2015. New Mexico Content Standards had recently been replaced with Common Core State Standards (CCSS). It was a busy year for APS.

## The Dilemma

How could APS Teachers deliver the CCSS, with fidelity, to English Language Learners (ELL) students, while ensuring compliance with the district goals; and at the same time, prepare students for a new state assessment that not only increased the cognitive complexity focus of the content, but would quickly require students to be assessed on-line.

## The Idea

When STEM Trajectory Initiative (STI) was launched in 2013, it was not a district initiative per-se, but instead, a non-traditional approach to the "delivery of instruction". Since most APS students are either identified ELL students or at a minimum, second language learners, it had been shown that these students needed hands-on experiences with multiple modality exposure to best learn the standards. Traditional paper-pencil, lecture, guided practice, Independent practice, sit-and-get teaching styles were lacking evidence of success.

Specific to mathematics, teachers needed to shift from focusing primarily on computational skills and symbol manipulation to a strategy that would build deep understanding of key mathematical ideas and promote the application of skills and concepts.

The founding partners of STI (Charles Newman, Valley HS; Mark Hendricks, West Mesa HS; and Alma Ripley, Carlos Rey ES), with the support of the VHS, WMHS and CRES Principals, all came to the same conclusion. To create curiosity and connections among teachers and students, it was necessary to apply everything

we knew about student success within the student demographics we were serving to the initiative. This initiative would be:

- Cost effective and sustainable.
- Of interest to New Mexico Public Education Department and Industry Leaders.
- Introduce STI STEM projects at the elementary level, preparing students with Project-Based Learning (PBL) opportunities that would "hook" them with the motivation to continue along a STEM track through middle school, high school, and college.
- A new and exciting, hands-on, real-world type approach for students to master the complex skills needed to be successful in the 21st century workforce.

We knew that this initiative would take 110% commitment, on-going planning, implementing, assessing and revising of our STI projects, and a lot of imagination, determination and gumption.

We also knew that an untapped resource was "knocking at our door". Radio amateurs have the passion to pass on this wonderful love of technology to future generations, as well as the desire to become "Life Long Learners". With a few presentations to the ham community about the need to support an already established curriculum, standards and strategies, they rallied 100% around STI.

The continued investment in this initiative is justified because teachers are still in transition from planning, instruction and assessing students "how they have always done it," to teaching abstract thinking and processing skills, necessary for students to be able to use those skills across all content areas, and develop into life-long learners with a mature sense of passion, desire, creativity and curiosity.

**Background of CubeSAT Project**

Over the past four years, STI has been successful beyond our original expectations. STI has touched the lives of students in at least 28 APS schools as well as another 12 schools from around New Mexico and Colorado. Within APS, students and STI staff have launched over 60 High Altitude Balloon (HAB) missions

which provided the opportunity for student payload (and CubeSAT subsystem) development innovation and preparation necessary to move STI into Phase 2: building a CubeSAT that will orbit the earth for several years and provide countless lessons for students and educators the world over.

## The CubeSAT Project

During SY2016-17, countless hours of research and collaboration was invested in creating a Framework and Syllabus for a Dual Credit (High School / College), Central New Mexico College (CNM) Electronics Course. In June 2017, Valley High School took the lead and hosted the first Dual Credit Electronics Summer STEM Course. 15 students from 3 APS high schools, completed the course and received 3 hours college credit from CNM, 1-hour elective credit from APS, and all passed the examination for their FCC Technician Class License in the Amateur Radio Service, plus 1 General Class License.

This course kicked off a 3-year effort to design, build, launch and operate an orbital CubeSAT. The CubeSAT multi-year project will involve current APS students, future APS students and former APS students that have entered universities, pursuing various STEM related degrees.

The Student Goals of the CubeSAT Project are:

- Mastery of 21st Century satellite technology using Project Based Learning.
- Define an achievable project to specify, design, implement, test and execute the construction of an actual CubeSAT.
- Plan a long-range project – utilizing Project Management skills for planning purposes.
- Develop technical and engineering skills
- Be exposed to top colleges and universities expressing interest in partnering with us on this multi-year project.
- Opportunities for civic engagement.
- Opportunity to “plan my future”.

The STI Goals of the CubeSAT Project are:

- Partner with AMSAT and industry leaders for technology and services to enhance the probability of success.
- Partner with New Mexico Technical Institute (NMT) with future Dual Credit course opportunities and potential high school summer internships.
- Serve as a foundational piece for Valley High School's newly earned title of, "APS STEM Magnet High School" for the district.
- Spend 2-3 years to design, build and test.
- Provide scientific data and experiences for all students over the next 5 – 7 years (2-3 prior to launch and 3-5 while in orbit).
- Provide video and infrared images of earth.
- Provide the capability to be used as a digital data and voice repeater, allowing anyone with an amateur radio license with line-of-sight visibility of the satellite to communicate through the satellite to distant stations.
- Provide a continuous source of scientific telemetry data, which can be received, and downloaded using a simple laptop based ground station, also being developed as part of this project.

CubeSAT Project Timeline:

- Summer, 2017: Cohort 1 Dual Electronics Course
    - Build Proof-of-Concept subsystems
        - Communications – Voice Repeater
            - Primary: Mode UV (UHF in/VHF Out)
            - Secondary: Mode VU (VHF in/UHF Out)
        - Photographic Imagery
            - Daylight Video Camera (432MHz ATV)

        - Daylight Still Camera
    - Telemetry and Control
        - Touchtone on UHF (70 cm)
        - APRS Data with GPS Position, Component Temperatures, Orientation (Pitch, Roll, Yaw)
        - AMSAT Telemetry using DUV (Data Under Voice)

- SY2017-18:
    - Weekly evening class to continue work on sub-systems
    - Test on HAB missions throughout the year, analyze, revise, re-test\
    - Begin Environmental Testing on Proof of Concept hardware to drive home the facts relating to the environment of space

- Summer, 2018:
    - Cohort 1 (advance into integrated systems, prototype and ground station assembly, and environmental testing)
    - Cohort 2 Dual Electronic course (identical Syllabus from Summer, 2017)

- Year 3:
    - Continue to bring younger students into the program
    - Advance older students to increasingly difficult tasks and licenses
    - Build and Qualify Production Configuration

- Year 4:  Launch and Operation

## Conclusion

This program started with getting students involved in High Altitude Balloon missions, evolved through payload development for the HAB missions, through an ISS contact involving a dozen APS schools, to last summer's dual credit electronics course where over 15 new hams spent 10 hours per day, 5 days per week for 3 weeks, plus Field Day weekend learning hands-on ham radio, electronics and physics. These students are now hooked on learning and STEM. Equally as impressive, the adult mentors from industry and the amateur radio community are now hooked on helping students. The school district and the APS Education Foundation are also hooked, as the STI has proven itself as a highly effective way to reach a lot of students for a minimal outlay of precious funds.

# The Radio Amateur Satellite Corporation

Joseph Spier
K6WAO
Executive Vice President

AMSAT®

NxtGen Crystal Radio

## NxtGen Crystal Radio

- AMSAT-NA Education Goals
  - Benefit Youth through STEM Education
  - Raise awareness through Educational Outreach
    - ARISS
    - AMSAT-Edu
  - Develop Partnerships with K-16 institutions for NASA's CubeSat Launch Initiative (CSLI) Program
  - Maintain CubeSat Simulators / Mechanical Mock up for Educators and Area Coordinators

- Keep Amateur Radio In Space

## NxtGen Crystal Radio

- Only one person responded that their Crystal Radio also transmitted (thank you Burns Fisher, W2BFJ)
- If I would have asked this question 50 years ago, I wonder what the answer would have been?
- It likely would have been –
- "I built a Crystal Radio Receiver" or "I built a Spark Gap Transmitter"

## NxtGen Crystal Radio

- So what is today's Crystal Radio Receiver?
- That depends on today's Youth and where their interests lay
  - But it better be Digital
  - One or more of what I call "The Big 3"
    - Robotics
    - Programming (like Ras Pi)
    - Space
  - And it better have an App

## NxtGen Crystal Radio

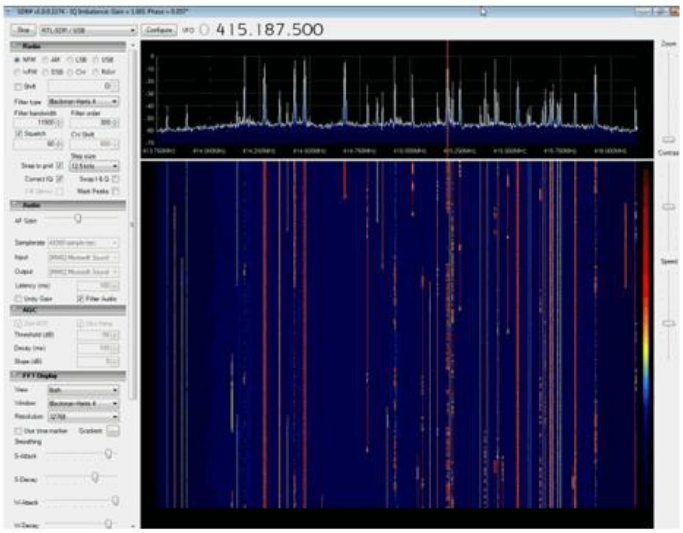
- So for AMSAT-NA Education that created a challenge
- The goal was to educate by “kit building” and have that benefit AMSAT and STEM educationa
- I also had a deadline
- HamVention 2017
  - Carole Perry’s ( WB2MGP) Instructor’s Forum
  - With a developed “Kit”

## NxtGen Crystal Radio

- So what is today’s Crystal Radio?
- The components would probably need to be:
- Software Defined Radio (SDR)
  - Digital
  - Programmable
- Antenna with Coaxial cable
  - Space (Tuned for Satellite Frequency Band)

# NxtGen Crystal Radio

- Software or the "App"
  - SDR
    - Digital
    - Programmable
  - FoxTelem
    - Digital
    - Programmable
    - Space

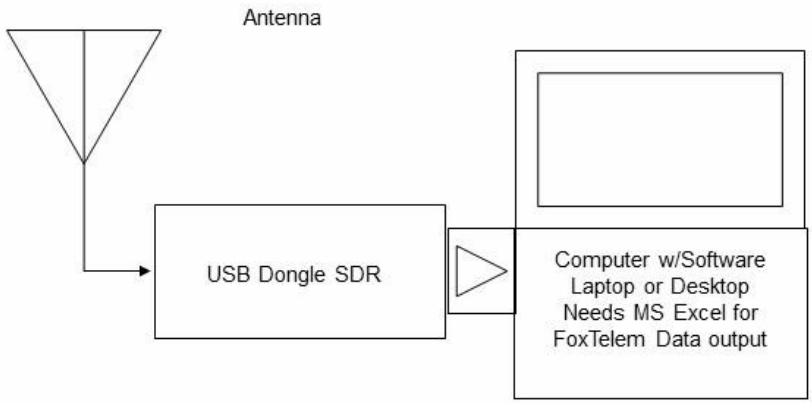



# NxtGen Crystal Radio

- Additional Parameters
  - Cost Efficient (Need to provide 10 – 12 kits)
  - Compact Size (must be classroom friendly)
  - Reasonable Performance
  - Unattended Configuration

- Beginning Assumptions
  - Hard Part
    - SDR Choice
  - Easy Part
    - Antenna Selection
    - Simple Software Porting of SDR to FoxTelem Program

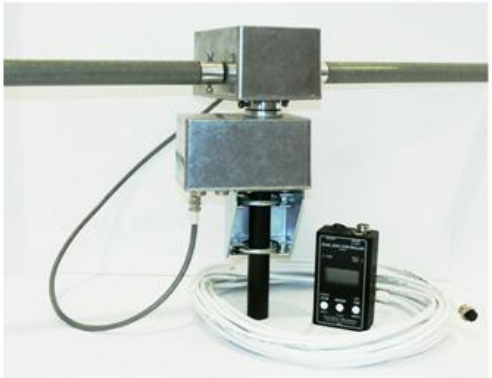
# NxtGen Crystal Radio




NxtGen Crystal Radio
Basic Design

# NxtGen Crystal Radio

- The Easy Part
- Antenna Selection
  - M² Antennas EB-144 Eggbeater
  - Quadrifilar Helicoidal
  - Arrow w/Rotator
  - Elk w/Rotator
  - Tape Measure w/Rotator
- Eliminated before Starting
  - Arrow w/Rotator
  - Elk w/Rotator
  - Tape Measure w/Rotator
- Additional Rotator/Cabling/Control Software Required
  - Cost Efficient – No (10 x $300(minimum) = $3000)



## NxtGen Crystal Radio

- M² Antennas EB-144 Eggbeater
  - 40” in diameter
  - 6 dBi Gain with reflector kit
  - List Price $273
- Benefits
  - Omni/Right Hand Circular Directional Antenna
  - Reasonable Performance
  - Unattended Configuration
- Disadvantages
  - Cost Efficient – No (10 x $273 = $2730)
  - Compact Size (must be classroom friendly) – Very Large

## NxtGen Crystal Radio

- Quadrifilar Helicoidal (QFH)
- Benefits
  - Omni Directional Antenna
  - Excellent Performance designed for satellites
  - Unattended Configuration
  - Cost Efficient
- Disadvantages
  - Compact Size (must be classroom friendly) – Long
    - 27.1” in Vertical Height
  - Semi-Complex Construction (Would need to be shipped Assembled) Balun Required
- Excellent Design Calculator at https://www.jcoppens.com/ant/qfh/calc.en.php

## NxtGen Crystal Radio

- Quadrifilar Helicoidal (QFH) consists of two loops
  - To obtain a circular polarization, it is necessary that the currents in both loops have a phase difference of 90 degrees. Though it is possible to make the phase difference with a phasing line (which is popular with the turnstile antenna), the most popular solution for the QFH is to use different loop sizes. This causes the current to be inductive in one, and capacitive in the other. Though the currents in both loops are reactive, connecting them in parallel causes the feed point, at the top, to be at nearly 50 Ohm, resistive.
  - The only problem remaining, is that the feed point, as with most antennas, is symmetrical. This means we shouldn't connect a coax cable (asymmetric) at this point. Though the antenna will be adapted, currents on the outside of the cable will offset the radiation pattern of the antenna.

## NxtGen Crystal Radio

- Quadrifilar Helicoidal (QFH) several feed point solutions exist:
  - **Transformer**
    - We could install a balun transformer at the top, but simple baluns transform impedance too - we want to maintain 50 Ohms.
  - **Choke**
    - Applying a couple of turns to the coax cable at the top will make the braid act as an RFC choke and block outside currents.
  - **Infinite balun**
    - Feeding the cable through one of the loop sides, will form an 'infinite balun'.
  - **Preamplifier**
    - Another - elegant - solution would be to install a preamplifier at the top, and design it for symmetrical input. As far as its know, none are on the market, so we'd have to make our own...

- Quadrifilar Helicoidal (QFH) Feed point Solutions Con't
  - Finally, using simply two loops, without the strange 'twist' which is so typical of the QFH, would generate a much more directional (vertical) radiation pattern. Also, the polarization would turn to linear near the horizon. The 'twist' generates the desired hemispherical pattern, which permits us to hear from horizon to horizon.
  - Choke Baluns
    - A simple solution for eliminating the exterior cable currents, is to wind the cable a number of times around the vertical support, making the outside act as an RF coke (inductor). The diameter should be as small as is practical, to reduce radiation as much as possible. The number of turns depends on the diameter, of course, but the inductance should be large enough (compared to 50 Ohm) to reduce the outside currents.
    - The windings should be as near to the top as possible, to reduce the length of the radiating parts

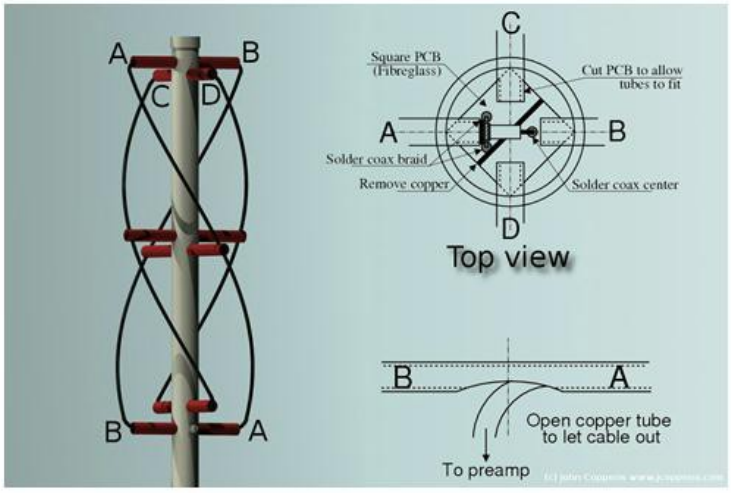
NxtGen Crystal Radio

- Quadrifilar Helicoidal (QFH) Feed point Solutions Con't
  - Infinite Baluns
    - Many questions arise about the infinite baluns... The impedance at the top is the one we want at the bottom too. So we cannot tolerate impedance transformation through the balun. This means we *have* to use a cable with the same impedance - else there would be a transformation!
    - For UHF and SHF work, the most practical solution is probably to use coax cable for the actual antenna itself. For lower frequencies, we should use tubing, and feed a 50 Ohm cable through it.
    - Part of the infinite balun principle, is the cable's braid should be connected to the antenna at the bottom center. In practice it has been difficult to detect much difference.
- Excellent QFH resources at
  https://www.jcoppens.com/ant/qfh/index.en.php

# NxtGen Crystal Radio

**Choke balun**

*ON7EQ's QFH with balun*




**Infinite balun**

# NxtGen Crystal Radio

- Quadrifilar Helicoidal (QFH) is an excellent additional project for NxtGen Crystal Radio
  - All QFH Resources from https://www.jcoppens.com/ant/qfh/index.en.php

## NxtGen Crystal Radio

- The Easy Part becomes the Hard Part
  - All antennas options were eliminated
- AMSAT website to the rescue
- Station and Operating Hints
  - https://www.amsat.org/station-and-operating-hints/
- An EZ-Lindenblad Antenna for 2 Meters
  - http://www.amsat.org/wordpress/wp-content/uploads/2015/08/An-EZ-Lindenblad-Antenna-for-2-Meters2.pdf
- Antenna developed for Fox CubeSats by Anthony Monteiro, AA2TX (SK)

## NxtGen Crystal Radio

- An EZ-Lindenblad Antenna for 2 Meters
  - Designed as an airport antenna in the 1930's
  - RHCP
  - Omni directional
  - With most of its gain at low elevation angles, it is ideal for accessing low earth orbit (LEO) Amateur Radio satellites.
  - Because it is omnidirectional, it does not need to be pointed at a satellite so it eliminates the complexity of an azimuth/elevation rotator system.
  - Meets Design Parameters
    - Cost Efficient
    - Compact Size (must be classroom friendly)
    - Reasonable Performance
    - Unattended Configuration

## NxtGen Crystal Radio

- This type of antenna was devised by Nils Lindenblad of the Radio Corporation of America (RCA) around 1940.[1]
  - His idea was to employ four dipoles spaced equally around a λ/3 diameter circle with each dipole canted 30° from the horizontal. The dipoles are all fed in phase and are fed equal power. The spacing and tilt angles of the dipoles create the desired antenna pattern when the signals are all combined.
    - *[1] G. Brown and O. Woodward, Jr., Circularly Polarized Omnidirectional Antenna, RCA Review, Vol 8, no. 2, Jun 1947, pp 259-269.*

## NxtGen Crystal Radio

- After World War II, George Brown and Oakley Woodward, also of RCA, were tasked with finding ways to reduce fading on ground-to-air radio links at airports.[2]
  - These links used linearly polarized antennas. The maneuverings of the airplanes often caused large signal dropouts if the antennas became cross-polarized. Brown and Woodward realized that using a circularly polarized antenna at the airport could reduce or eliminate this fading so they decided to try Lindenblad's TV antenna concept.
    - *[2] G. Brown and O. Woodward, Jr., Circularly Polarized Omnidirectional Antenna, RCA Review, Vol 8, no. 2, Jun 1947, pp 259-269.*

## NxtGen Crystal Radio

- Brown and Woodward designed their antenna using metal tubing for the dipole elements.
  - Each dipole element is attached to a section of shorted open-wire-line, also made from tubing, which serves as a balun transformer. A coaxial cable runs through one side of each open-wire-line to feed each dipole. The four coaxial feed cables meet at a center hub section where they are connected in parallel to provide a four-way, in-phase power-splitting function. This cable junction is connected to another section of coaxial cable that serves as an impedance matching section to get a good match to 50 Ω.

## NxtGen Crystal Radio

- While the Brown and Woodward design is clever and worked well, it would be quite difficult for the average ham to duplicate.
  - The major cause of the difficulty in designing and constructing Lindenblad antennas is the need for the four-way, in-phase, power splitting function. Generally we want to use 50 Ω coaxial cable to feed the antenna, we have to somehow provide an impedance match from the 50 Ω unbalanced coax to the four 75 Ω balanced dipole loads.
  - Previous designs have used combinations of folded dipoles, open-wire lines, twin-lead feeds, balun transformers and special impedance matching cables in order to try to get a good match to 50 Ω. These in turn increase the complexity and difficulty of the construction.

- The EZ-Lindenblad
  - An antidote to these difficulties is the EZ-Lindenblad.
  - The key concept of the EZ approach is to eliminate anything that is electrically or mechanically difficult, leaving only things that are easy. This leads to the idea of just feeding the four dipoles with coax cable and soldering the cables to a connector with no impedance matching devices at all. This would certainly be easy but we also want the antenna to work! Without the extra impedance matching devices, how is it possible to get a good match to 50 Ω?

NxtGen Crystal Radio

- The EZ-Lindenblad
  - If we could get each of the four coax feed cables to look like 200 Ω at the connector, then the four in parallel would provide a perfect match to 50 Ω.
    - We could do this if we used quarter-wavelength sections of122 Ω coax to convert each 75 Ω dipole load to 200 Ω. Unfortunately, there is no such coax that is readily available.
    - But we can accomplish the same thing with ordinary 75 Ω, RG-59 TV type coax if we run the cable with an intentional impedance mismatch. By forcing the standing wave ratio (SWR) on the cable to be equal to 200/75, or about 2.7:1, we can make each cable look exactly like 200 Ω at the connector as long as we make them the right length.

- The EZ-Lindenblad
  - It is easy to make the SWR equal 2.7:1 by just making the dipoles a little too short for resonance.
    - An EZNEC antenna model can be used to determine the exact dipole dimensions. [3]
    - The conversion from the balanced dipole load to the unbalanced coax cable can be painlessly accomplished by threading each cable through an inexpensive and readily available ferrite sleeve making essentially a choke balun.
    - The only remaining issue is the required length of the feed cables, and this can be easily determined using a Smith Chart.
    - [3] EZNEC+ V4 antenna analysis software by Roy W. Lewallen, W7EL, available from www.eznec.com.

NxtGen Crystal Radio

- The required length of 75 Ω coax to provide a 200Ω load was determined with a Smith Chart.
  - An EZNEC antenna model was used to simulate cutting the dipole lengths until the SWR on the line reached 2.7:1. The model showed that the dipole load impedance would then be 49 – j55 Ω.
    - The length of the coaxial line calculated to 0.374 λ.
  - Designing for a center frequency of 145.900 MHz, optimizes performance in the satellite sub band.
    - At 145.9 MHz, a wavelength is about 81 inches and since the coax used has a velocity factor of 0.78, we need to make the feed cables 81 × 0.374 × 0.78 = 23.6 inches long.

## NxtGen Crystal Radio

- EZ-Lindenblad Radiation Pattern favors the lower elevation angles.
  - The –3 dB points are at 5° and 25° with the maximum gain of 4.8 dBic (with respect to an isotropic circularly polarized antenna) at around 13° .
    - This is an excellent pattern for accessing LEO satellites.
    - Most of the satellite pass elevations will be in this range and it is also the elevation at which the satellite provides the best chance for DX contacts.
  - Higher elevation passes do not require as much gain.

## NxtGen Crystal Radio

- Construction of the EZ-Lindenblad was basic.
  - Cut the aluminum tubing to length.
  - Mount the cross boom tubing on the angle bracket.
    - After constructing the prototype, the angle bracket (mounting pole location) was centered by slightly offsetting the cross boom mounting holes. Flat head bolts were also used to keep angle bracket flat against the mounting pole.
  - Assemble and mount the dipoles on the cross booms at 30º angle.
  - Cut, trim, assemble, and solder RG-59 coax into a phasing harness with N connector and ferrite choke.

# NxtGen Crystal Radio

Various Construction Photos of EZ-Lindenblad Antenna

NxtGen Crystal Radio

- The EZ-Lindenblad
  - The assembled antenna measures about 15” in height and 26” in width. To ship 12 assembled antennas would be approximately two shipping pallets by truck freight.
  - An Engineering exercise in packaging began.
    - The antennas were mechanically disassembled. The cross booms, assembled dipoles, and angle brackets were fitted into 36” long x 3” diameter mailing tubes. Colored dots were used to ease in re-assembly.
    - All bolts, screws, and phasing harness were placed into zip lock bags and instructions to assemble were put into a small box with the rest of the NxtGen Crystal Radio kit.

# NxtGen Crystal Radio

Assembly / Packaging Photographs

# NxtGen Crystal Radio

- The EZ-Lindenblad
- Test Results

Packaging Results

# NxtGen Crystal Radio

Assembled EZ-Lindenblad Antenna at Hamvention 2017

NxtGen Crystal Radio

- The Hard Part becomes the Easy Part
  - Remember those Beginning Assumptions
    - Hard Part
      - SDR Choice
    - Easy Part
      - Antenna Selection (HARD PART)
      - Simple Software Porting of SDR to FoxTelem Program

- Software Defined Radio Choices
  - RTL-SDR $20
  - SDRPlay RSP1 $90
  - FunCube Pro+ Dongle £124.99 or $163

## NxtGen Crystal Radio

- The Hard Part becomes the Easy Part
  - Eliminated before Starting
    - FunCube Pro+ Dongle £124.99 or $163
    - Cost Efficient – No (10 x $163(minimum) = $1630)
  - The FunCube Pro+ is an excellent SDR for FoxTelem.
    - A major benefit is the Fox Telem software is developed to recognize the FunCube Pro+ dongle as an input.

## NxtGen Crystal Radio

- SDRPlay RSP1 $90
- $90
- Benefits
  - Cost Efficient
  - Unattended Configuration
  - Reasonable Performance
  - Low noise floor
  - 10kHz to 2 GHz
  - FM, TV, WeFAX,etc.
- Disadvantages
  - Compact Size – Requires USB-B Cable

## NxtGen Crystal Radio

- RTL-SDR
  - $20
  - Benefits
    - Cost Efficient
    - Compact Size
    - Unattended Configuration
    - 52 MHz – 2200 MHz
    - FM, TV, WeFAX,etc.
  - Disadvantages
    - Reasonable Performance – Marginal
    - Noise Floor ~-60dB

## NxtGen Crystal Radio

- Methods to improve RTL-SDR performance
- JaniLabs RTL-SDR Pre-Amplifier
  - Operating Frequency: 1MHz to 3GHz
  - RF Gain adjustable: -20dB to +19dB
  - Noise: 1.9-2.1dB
  - Connectors: SMA Female Jack

## NxtGen Crystal Radio

- The Cost Analysis
  - RTL-SDR + SMA connector + JaniLabs Pre-Amp + Power Supply
    - $20 + $5 + $20 + $20 = $65
  - SDRPlay RSP1 + USB-B cable
    - $90 + $5 = $95
  - FunCube Dongle Pro+
    - $163
- RTL-SDR selected based on lowest cost, moderate performance, and easy availability

## NxtGen Crystal Radio

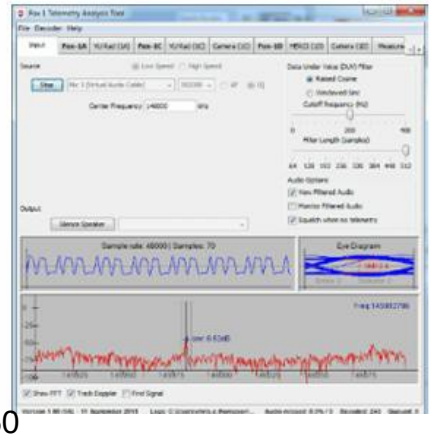
- The Easy Part becomes the Hard Part (Part 2)
  - Remember those Beginning Assumptions
    - Hard Part
      - SDR Choice (EASY PART)
    - Easy Part
      - Antenna Selection (HARD PART)
      - Simple Software Porting of SDR to FoxTelem Program
- The "App"
  - SDR Software
  - FoxTelem Soltware

## NxtGen Crystal Radio

- The FoxTelem Software
  - Download from AMSAT-NA website
    - https://www.amsat.org/foxtelem-software-for-windows-mac-linux/
  - We get some ideas on SDR configuration from the instructions on using the FunCube Pro+ Dongle
  - A software defined radio is a better choice because we get no filtering of the low frequency audio. The Funcube Dongle, for example, can receive the telemetry and pass the IQ baseband audio

## NxtGen Crystal Radio

- You have two choices to set this up.
  - Either receive the audio in your favorite SDR software and feed the demodulated FM audio to the telemetry decoder
  - Or use the IQ demodulation in the telemetry decoder itself.
  - Both should work.



## NxtGen Crystal Radio

- Here are the advantages of using FoxTelem's IQ demodulator:
  - Advantages
    - FoxTelem will find the satellites signal when it appears and will follow the Doppler. You do not need a separate application to set the frequency assuming it is in the pass band of the SDR.
    - FoxTelem will measure the parameters about the received signal such as Signal to noise ratio (SNR) and Time of Closest Approach (TCA), which AMSAT would like to collect and analyze.
    - You don't need to use another piece of software, such as Virtual Audio Cable, to connect the two programs. Or feed the audio out of the computer and back in again with a physical cable.

## NxtGen Crystal Radio

- Here are the disadvantages of using FoxTelem's IQ demodulator:
  - Disadvantages
    - A separate SDR program may have better capabilities than the SDR in the Telemetry Decoder.
    - Your existing SDR may already be configured to work with Doppler correction software and the rest of your ground station
    - FoxTelem can be more compute-intensive than some other SDR programs and may not work as well on slower computers
  - Remember all these caveats are for a selected input in the FoxTelem software.
- How do we connect a wide band SDR?

## NxtGen Crystal Radio

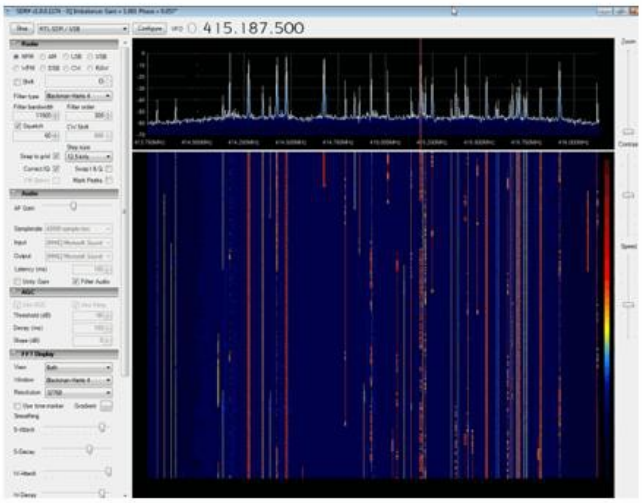
- In the past, we simply would have made an interface cable.
  - Radio to decoder, need an adaptor cable
- The same is true today only Digital
  - The RTL-SDR and SDRPlay are receivers with a digital interface software for tuning.
  - The RTL-SDR also needs some software configurations
  - This is accomplished by using SDR# software.
- There are some very good RTL-SDR and SDR# YouTube videos – just Google them!

## NxtGen Crystal Radio

- SDR# (Radio) + ? (Cable) + FoxTelem (Decoder)
  - What is used for the cable?
  - Two Choices
    - Virtual Audio Cable Software
      - http://software.muzychenko.net/eng/vac.htm
    - VB Audio Cable
      - https://www.vb-audio.com/Cable/index.htm
  - Both are fairly system specific so configurations are customized.

## NxtGen Crystal Radio

- SDR# + VAC + FoxTelem = “App”
  - SDR#
    - Receiver Software
  - Virtual Audio Cable or VB Audio Cable
    - Takes the output from SDR#
    - Feeds to the input of FoxTelem
  - FOXTelem Software
    - Data Under Voice (>300Hz) Telemetry for all Fox




## NxtGen Crystal Radio

- The Easy Part becomes the Hard Part (Part 2)
  - Remember those Beginning Assumptions, again
  - Here’s the Reality Check
    - Hard Part
      - SDR Choice (EASY PART)
    - Easy Part
      - Antenna Selection (HARD PART)
      - Simple Software Porting of SDR to FoxTelem Program (EASY/HARD PART)

## NxtGen Crystal Radio

- AMSAT Satellite Ground Receiving Station
  - RTL-SDR
  - JaniLabs RTL-SDR PreAMP
  - EZ-Lindenblad Antenna
  - FOXTelem Software
  - Total Cost = $180 each

## NxtGen Crystal Radio

- Future Experimentation
  - Antenna Designs
    - Quadrifilar Helicoidal
    - 70cm Parasitic Lindenblad Omni Antenna
    - http://www.amsat.org/wordpress/wp-content/uploads/2014/01/70ParaLindy.pdf
- EXCEL Data to NASA GMAT
  - Real Mission Data ported to a Graphic Model
    - http://gmatcentral.org/

# NxtGen Crystal Radio

- Future Experimentation
  - LimeSDR $299
    - LimeSDR is an open source, apps-enabled software-defined radio (SDR) platform that can be used to support just about any type of wireless communication standard; it can transmit and receive UMTS, LTE, GSM, LoRa, Bluetooth, Zigbee, RFID, and Digital Broadcasting
  - https://myriadrf.org/projects/limesdr/

## For More Information
## WWW.AMSAT.ORG

AMSAT-North America
10605 Concord Ave. Suite 304
Kensington, MD 20895

tel. (301) 822-4376
(888) 322-6728 toll free

e-mail: MARTHA@AMSAT.ORG

# Questions?

AMSAT-NA
Radio Amateur Satellite Corporation

Joe Spier, K6WAO
Executive Vice President
16745 Mount Rose Hwy
Reno, NV 89511

(530)305-0702
k6wao@amsat.org
http:/www.amsat.org

# AMSAT Engineering Documentation Update

**Compiled by Jerry Buxton, N0JY**
**AMSAT Vice President - Engineering**
**n0jy@amsat.org**

AMSAT, as an educational organization, would like to publicly release the majority of our design documentation to serve as a learning tool to anyone interested in satellite development. However, in order to avoid complications with ITAR and Export Administration Rules, the information must first be released via an openly available publication. We would also like to be able to discuss our satellite projects with our own members, some of whom are not "US-persons" per those regulations. These AMSAT Space Symposium proceedings provide a convenient mechanism for the needed publication in order to make this information public domain and allow us to communicate with our members.

Fox-1 and other Engineering documents related to our satellites have been published in previous *Proceedings*, and here we continue with RadFxSat-2/Fox-1E documents as well as some Fox-1 mechanical drawings. The publishing of this information also makes it available to anyone who wishes to use the designs or design ideas for their own satellites, hopefully with the intention of making more amateur radio satellites available to the world.

**Date:** October 30, 2016
**Version:** Version 1.02

# Fox-1E
(RadFxSat 2)
# System Requirements Specification

## 1 Introduction

This document specifies the system level requirements for the AMSAT Fox-1E satellite project. This 1-Unit CubeSat is one part of the AMSAT Fox program and includes a subset of the capabilities to be realized in the overall program.

Fox-1E is intended as a follow-on and upgrade to the initial Fox-1 series CubeSat satellites. The first four Fox-1 satellites flew single channel FM repeaters as their bus-supporting communications system. (Fox-1E will have essentially the same configuration and basic bus structure as the earlier Fox-1 satellites, but with their single channel FM repeater transmitter and receiver replaced with a new wideband linear transponder. This type of amateur radio communications system will provide more channels for multiple, simultaneous radio contacts, as well as a higher data rate, continuous telemetry downlink.

In addition to its mission as a communications satellite, Fox-1E will host an experiment payload. The satellite will reserve mass and volume for the experiment and will provide DC power and communications capability to support experiment data downlinking. The (radiation) experiment will be provided by students at Vanderbilt University's Institute for Space and Defense Electronics (ISDE).

### 1.1 Document History

| DATE | VERSION | AUTHOR | SUMMARY |
|---|---|---|---|
| November 4, 2015 | 1.0 | J. Buxton (N0JY) | Original Fox-1 SRS |
| April 4, 2016 | 1.01 | E. Skoog (K1TVV) | Update draft for review/comment |
| October 30, 2016 | 1.02 | E. Skoog (K1TVV) | Rqmt clarifications and updates |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## 1.2 Document Scope

The purpose of this document is to specify the requirements of the satellite at the system (i.e., "black box") level. It is intended to be used by hardware, software and mechanical designers to develop architecture/high-level design specifications. It is also intended to be used for test planning and development.

## 1.3 Document Format

This document provides requirements in numbered format. Each requirement is assigned a unique number. Additional information such as comments or examples provided for guidance or clarity is *italicized* to distinguish it from specific requirements.

## 1.4 References

1. AMSAT Fox-1 Concept of Operations, Version 1.03, October 19, 2011.
2. CubeSat Design Specification, Rev. 13, February 20, 2014, The CubeSat Program Cal Poly SLO.
3. Launch Services Program, Program Level Dispenser and CubeSat Requirements Document, LSP-REQ-317.01 Revision B, January 30, 2014, National Aeronautics and Space Administration (NASA).
4. FANTM-RAIL – Railed 3U++ Single (FRA-R3S) Attachment/Deployment Assembly Interface Description Document (IDD), FRA-IDD-001, Revision A, Xtenti, 25 October 2016.
5. ITU Radio Regulations, Edition of 2012, available from http://www.itu.int/publ/R-REG-RR-2012/en

## General Requirements

### 1.5 CubeSat Requirements

1.5.1 The satellite shall satisfy the requirements specified in Xtenti FRA-R3S FANTM-RAIL – Railed 3U++ Single Attachment/Deployment Assembly FRA-IDD-001, Rev A.

1.5.2 The satellite shall satisfy the requirements specified in the CubeSat Design Specification, Rev. 13.

1.5.3 The satellite shall satisfy the requirements specified in the NASA LSP-REQ-317.01 Revision B.

1.5.4 The satellite shall satisfy the requirements for a 1 Unit (single) CubeSat.

1.5.5 The satellite shall provide mass for an experiment payload up to 100 g.

1.5.6 The satellite shall provide volume for an experiment payload up to 95 x 95 x 31.4 mm.

### 1.6 Environmental Requirements

1.6.1 The satellite avionics shall be designed for an operating temperature range of $-40^0$C to $+85^0$C.

1.6.2 The satellite shall be designed to operate in an approximate 500 to 700 km, sun-synchronous, circular orbit.

1.6.3 The satellite shall be designed to tolerate the radiation environment in orbit.

### 1.7 Reliability Requirements

1.7.1 The satellite shall be designed for a minimum 5-year, on-orbit lifetime.

### 1.8 Radio Frequency (RF) Regulatory Requirements

1.8.1 The satellite's RF transmitter shall meet or exceed the requirements specified in ITU Radio Regulations, Volume 1 Articles, Chapter 1 – Terminology and Technical Characteristics of Stations, Article 3.

1.8.2 All satellite uplinks shall be in the 2 meter band of the Amateur Satellite Service.

1.8.3 All satellite downlinks shall be in the 70 cm band of the Amateur Satellite Service.

1.8.4 All satellite transmitter and receiver generated frequencies shall deviate by no more than 5 parts-per-million from their specified values including initial accuracy and temperature variation.

1.8.5 All satellite frequencies shall be coordinated with the IARU.

1.8.6 The command uplink shall be a narrow band FM signal (+/- 5 kHz deviation) located outside of the user transponder passband.
*The band plan with the actual coordinated frequencies will be specified in a separate document.*

# 2 Functional Requirements

## 2.1 Antenna System

2.1.1 The satellite shall include a deployable antenna system.

## 2.2 Attitude Control

2.2.1 The satellite shall incorporate passive magnetic stabilization to align the deployed antennas with the magnetic field of the earth.

## 2.3 Access Ports

2.3.1 The satellite shall include a "Remove Before Flight" (RBF) pin as per the CubeSat Design Specification.

2.3.2 The satellite shall include an umbilical port as per the CubeSat Design Specification.

## 2.4 Pre-launch Features

2.4.1 The satellite battery and photovoltaic panels shall be electronically disconnected from the avionics when the "Remove Before Flight" pin is inserted, regardless of the state of the deployment switch(es).

2.4.2 The satellite shall provide a means to charge the battery via the umbilical port while integrated with the dispenser.

2.4.3 The satellite shall provide a means to run diagnostic tests via the umbilical port while integrated with the dispenser.

## 2.5 Power

2.5.1 The satellite shall produce electrical power from sunlight.

2.5.2 The satellite shall produce electrical power while in sunlight regardless of orientation and while tumbling or spinning.

2.5.3 The satellite shall produce sufficient average electrical power to operate continuously in the orbit of maximum eclipse.

2.5.4 The satellite shall provide sufficient battery capacity to operate continuously in the orbit of maximum eclipse.

2.5.5 The satellite shall not provide battery power to the main bus until approximately 115 seconds after activation of both deployment switches indicating successful launch vehicle dispenser separation.

## 2.6 Experiment

2.6.1 The satellite shall provide DC power for an experiment payload.

2.6.2 The satellite shall provide a means to activate and deactivate the experiment payload.

2.6.3 The satellite shall provide a means to telemeter data from the experiment payload.
*The experiment payload details will be specified in a separate document.*

## 2.7 RF Uplink

2.7.1 The satellite shall include a linear uplink receiver.

2.7.2 The uplink receiver shall process signals in a linear manner. The receiver will process SSB, CW, PSK, and FSK signals to its intermediate frequency with intermodulation distortion (IMD) products no higher than -50 dBc when the maximum intended level signals are applied to the spacecraft.

## 2.8 RF Downlink

2.8.1 The satellite shall include a linear downlink transmitter.

2.8.2 The downlink transmitter shall be a linear transmitter with intermodulation products no higher than -20 dBc when the highest level output signal is at maximum output power.

## 2.9 Transponder

2.9.1 The satellite shall provide linear transponder operation via the RF uplink and RF downlink.

2.9.2 The user transponder shall have a -3dB bandwidth of 30 kHz and a -20dB bandwidth of 50 kHz, referenced to the center of the passband.

2.9.3 The transponder shall be an inverting transponder.

2.9.4 The transponder shall have a means of sensing and adjusting gain so that the highest level throughput signal is not distorted with intermodulation products higher than -20 dBc with multiple signals in the passband.

2.9.5 The transponder gain adjustment shall have a time constant of <TBD> seconds, so that the highest level signal in the passband will not 'pump' other signals in the passband by its modulation.

2.9.6 The transponder shall have a mechanism for detecting the level of signals in the uplink passband. This aggregate level shall have a threshold switching mechanism for deactivating the satellite transmitter when no signals exceed the threshold. The transmitter shall be activated when the threshold is exceeded for approximately one second. The threshold

mechanism shall continue to keep the transmitter activated for 30 seconds after the threshold conditions are no longer satisfied.

2.9.7 The transponder shall have a linear gain such that a received CW signal of <TBD> dBm will result in a power output of +20 dBm. Any other signals amplified in the passband shall create IMD products not to exceed -20 dBc in the passband of the transponder.

## 2.10 Telemetry Data

2.10.1 The satellite shall collect telemetry data.

2.10.2 The telemetry data shall include at a minimum, the measured parameters shown in Table 1.

**Table 1**

| PARAMETER NAME | DESCRIPTION |
|---|---|
| CELL V | Voltages of battery cells |
| PANEL V | Voltages of solar panels |
| TOTAL I | Total DC current out of power system |
| PA I | DC current into RF power amp |
| BATTERY T | Temperature of battery |
| PANEL T | Temperatures of solar panels |
| TX T | Temperature of RF transmitter card |
| RX T | Temperature of RF receiver card |

2.10.3 The measured parameters shall be sampled at least every 15 seconds.

2.10.4 The minimum and maximum values of each of the measured parameters shall be saved in non-volatile memory.

2.10.5 The telemetry data shall also include at a minimum, the calculated parameters shown in Table 2.

**Table 2**

| PARAMETER NAME | DESCRIPTION |
|---|---|
| UP TIME | Total seconds since avionics power-up or the last Reset |
| SPIN | Satellite spin rate and direction |

2.10.6 A telemetry frame shall include the current measured values, the saved minimum and maximum values, and the current calculated values.
*The detailed telemetry interface will be specified in a separate document.*

## 2.11 Telemetry Transmission

2.11.1 The satellite shall send slow speed telemetry using BPSK on the RF downlink.

2.11.2 The downlink telemetry signal shall be a 1200 bps BPSK signal located outside the user transponder passband and shall be part of the frequency multiplexed downlink. The telemetry signal's peak power level shall be no greater than <TBD> dB below the maximum output power of the transponder.

2.11.3 Telemetry data shall be transmitted simultaneously with user transponder signals, and shall cease to be transmitted when the user transponder threshold signal "hang time' of 30 seconds has expired.

2.11.4 The telemetry transmission shall include telemetry frames.

2.11.5 The telemetry transmission shall also include experiment data.

## 2.12 Command Capability

2.12.1 The satellite shall provide a means to process commands sent via the RF uplink from authorized ground control stations.

2.12.2 The command receiver shall be on a frequency outside the transponder bandwidth.

2.12.3 A 1200 bps AFSK-FM via a NBFM channel demodulation capability shall be provided to the IHU for the purpose of transferring data and commands to the IHU. This AFSK-FM channel shall follow the FSK requirements of Bell 202 modems.

2.12.4 The uplink command receiver shall be capable of receiving and demodulating AFSK-FM signals at 1200 bps with an input signal level of -120 dBm with a baseband Bit Error Rate (BER) of 1 x $10^{-4}$.

2.12.5 The command uplink receiver will not be responsible for bit or byte synchronization, re-timing, or error correction.

2.12.6 The following commands shall be provided, as shown in Table 3, below.

**Table 3**

| Command | Operation |
|---|---|
| SAFE MODE | Enter Safe Mode |
| INHIBIT TX | Inhibit RF transmission |
| ENABLE TX | Enable RF transmission |
| IHU OFF | Power off IHU |
| IHU ON | Power on IHU |
| CLEAR | Clear stored telemetry |
| TRANSPONDER MODE | Enter Transponder Mode |
| ENABLE AUTO-SAFE | Enable Auto-Safe Mode |
| DISABLE AUTO-SAFE | Disable Auto-Safe Mode |

2.12.7 A SAFE MODE command shall cause the satellite to enter the Safe Mode.

2.12.8 An INHIBIT TX command shall disable the RF transmitter.

2.12.9 An ENABLE TX command shall enable the RF transmitter.

2.12.10 An IHU OFF command shall cause the IHU System to power off.

2.12.11 An IHU ON command shall cause the IHU System to power on.

2.12.12 A CLEAR command shall cause the satellite to clear the saved minimum and maximum telemetry parameter values.

2.12.13 A TRANSPONDER MODE command shall cause the satellite to enter the Transponder Mode.

2.12.14 An ENABLE AUTO-SAFE command shall enable the auto-safe mode state.

2.12.15 A DISABLE AUTO-SAFE command shall disable the auto-safe mode state.

*The control interface details will be specified in a separate document.*

## 2.13 On-Orbit Operating Modes

2.13.1 The satellite shall provide on-orbit operating modes as shown in Table 4.

**Table 4**

| NAME | DESCRIPTION |
|---|---|
| Startup Mode | Wait 50+ minutes and deploy antennas |
| Safe Mode | Wait 120 seconds then begin telemetry beacon sequencing;<br>Experiment powered off |
| Auto-Safe Mode | If enabled, automatically detects a low battery condition and shuts down some major satellite activities |
| Transponder Mode | Linear transponder mode; telemetry channel and experiment active |

2.13.2 The satellite shall transition between its on-orbit operational modes as shown in the State Transition Diagram in Figure 1.

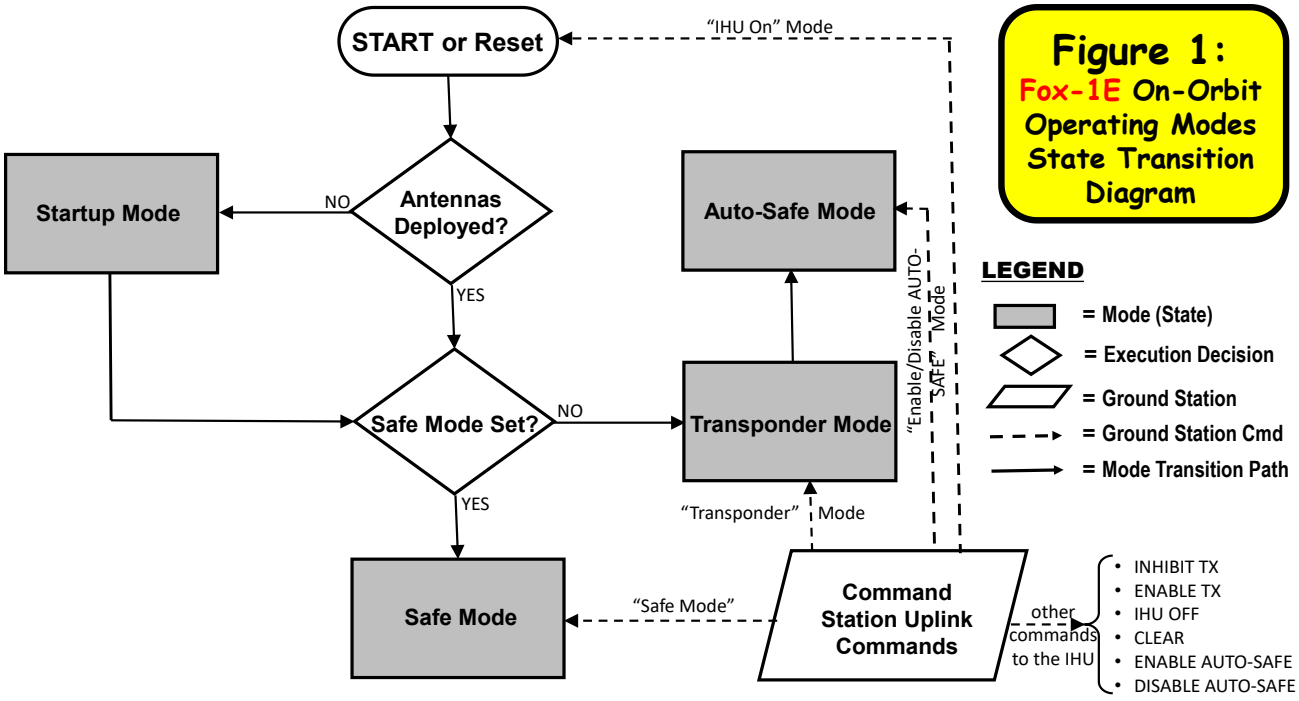



**Figure 1:** Fox-1E On-Orbit Operating Modes State Transition Diagram

2.13.3 Upon launch, a power cycle of the avionics, or after a reset action, the satellite shall enter the "START" point as shown in Figure 1.

2.13.4 An IHU ON command shall also cause the satellite to begin operation from the "START" point as shown in Figure 1.

2.13.5 If the antennas have not been deployed, the satellite shall enter Startup Mode.

2.13.5.1 In Startup Mode, the satellite shall wait at least 50 minutes before commanding antenna deployment and initiating RF transmissions.

2.13.5.2 After the antennas have been successfully deployed upon initial orbit, the satellite shall enter Safe Mode.

2.13.6 After a Reset action and antennas successfully deployed check, the satellite shall determine whether its last state was SAFE MODE.

2.13.6.1 If the last state was SAFE MODE, the satellite shall (re)enter SAFE MODE.

2.13.6.2 1200 bps BPSK telemetry beacon downlink channel operation shall occur during SAFE MODE. There shall be NO CW beacon functionality implemented in the satellite

2.13.6.3 If the last state was not SAFE MODE, the satellite shall enter TRANSPONDER MODE.

2.13.6.4 In TRANSPONDER MODE, the transponder and the telemetry shall be active.

2.13.7 The command RF uplink shall be monitored for commands in all modes.

### 2.14 Operational Timing Requirements

2.14.1 The satellite shall satisfy the operational times shown in Table 5.

**Table 5**

| FUNCTION | TIME (± 5%) | OPERATIIONAL CONDITIONS |
|---|---|---|
| Burn Resistor Activation | ≈ 4 seconds | Time allotted for the (activated) burn resistors' initial melting of the deployable antenna restrainers |
| Burn Resistors RE-activation | ≈ 20 seconds | Time allotted for the (RE-activated) burn resistors' subsequent melting of the deployable antenna restrainers |
| Deployment Switch Notification Delay | ≈ 115 seconds | Delay time from deployment switch physical activation to electronic activation report. |
| IHU Boot-Up/POST | < 20 seconds | Time for the Internal Housekeeping Unit (IHU) to boot-up and run Power On System Tests (POST) |
| Antenna/RF Initiation Delay | ≈ 50 minutes | Antenna deployment and transmission inhibit time after dispenser launch |
| Beacon (Safe Mode) Duty Cycle | ≈ 2 minutes | Elapsed time from end of beacon telemetry transmission to start of next beacon transmission while in SAFE MODE |
| Hang Timer | ≈ 30 seconds | RF transmit carrier time after the uplink passband (aggregate) signal(s) no longer satisfy the threshold criteria. |
| Telemetry Period | ≈ 15 seconds | Sampling period for all telemetry parameters |
| ENABLE TX Confirm | ≈ 3 minutes | Time after receipt of ENABLE TX command before transmitting a telemetry beacon |
| Watchdog Timer | ≈ 10 seconds | If IHU-directed tasks do not report back within this time period, the satellite will reset. |

## 3 External Interface Documents

To fully specify the satellite's technical requirements, the following documents should be consulted:

1. IARU Coordinated Frequency Plan
2. Downlink Specification
3. Control Interface Specification
4. Experiment Payload Specification

## 4 Summary

The *Fox-1E* satellite will be AMSAT's first linear transponder CubeSat. Its primary mission is to provide linear transponder communications capability. The secondary mission is to host a university-provided experiment payload.

# Fox-1E Linear Transponder Communications Performance Analysis

**OVERVIEW** – AMSAT-NA's Fox-1E (RadFXSat-2) CubeSat will feature the first linear transponding capability of the Fox-1x family of satellites. The Linear Transponder will support simultaneous, multi-channel communications using primarily CW and SSB modes. The design of the transponder will include state-of-the-art integrated circuit components to provide exceptional communications performance, while minimizing size, weight and power. Additionally, unlike previous Fox-1x CubeSats, Fox-1E will be a V/U satellite. This means that operator uplink transmissions will require generation of VHF (2 meter) SSB/CW amateur radio signals and downlink signal reception will be via UHF (70 cm) SSB/CW signals. A separate 1200 bps BPSK modulated channel will be dedicated to downlinking CubeSat telemetry and experiment data.

This paper includes details of the design analysis performed to develop a Fox Linear Transponder, specifically the architecture block diagram; initial component identification, and signal chain cascade analysis (including spurious products) of the critical performance parameters of the design.

**ARCHITECTURE** – There are many definitions of Architecture. One of the most widely used originates from the Institute of Electrical and Electronic Engineers (IEEE).

**ARCHITECTURE ≡ the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.**

Architectures are intended to be "fit for purpose." Consequently, there are myriad categories of <adjective> architectures, e.g., <enterprise>, <system>, <technical>, <operational>, <program>, <information>, <software>, <hardware>, <…etc...>. Regardless of the type of architecture, its constituent components are characterized by their capabilities, connections, and constraints. Specifically, in Radio Frequency (RF) electronics design, a <system> architecture is often referred to as an <RF> architecture. RF architectures are generally introduced early in the development cycle through a conceptual design created to satisfy the requirements using behavioral models or functional blocks. As such, the RF architecture represents the foundation of the entire RF design.

Fox-1E Linear Transponder initial design activities exemplify an early stage RF architecture intended to answer the following questions:

1. How many stages should be used?
2. What are the different types of stages required to satisfy the performance requirements?
3. What are the performance parameters of each stage?
4. What is the functional allocation for the stages' order/arrangement?

Top level RF architectures generally take the form of block diagrams of the major subsystems of the intended design. This is accomplished in order to evaluate preliminary requirements allocation, and functional decomposition, i.e., the scope of the subsystems. At the conceptual level, the Fox-1E Linear Transponder can be envisioned as a set of major subsystem functional blocks as illustrated in Figure 1 below.

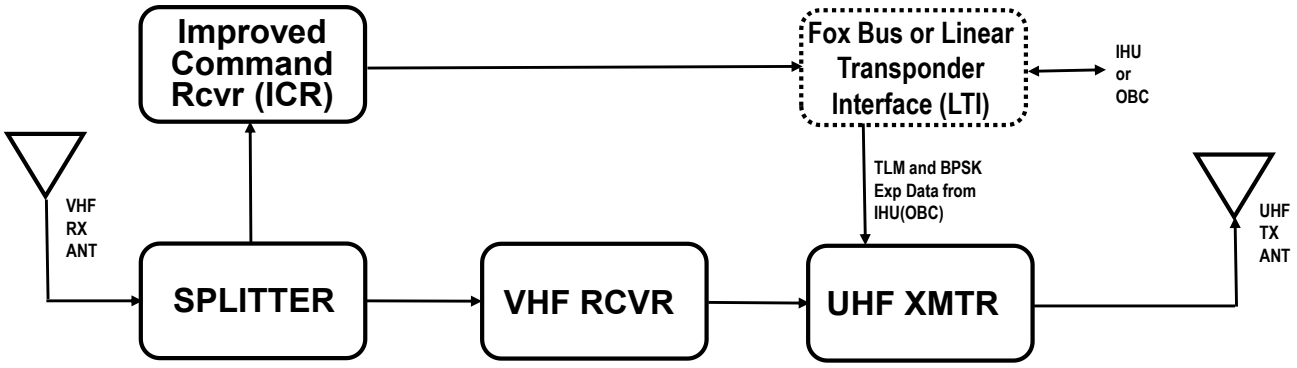



**Figure 1: Fox-1E Linear Transponder Top Level Block Diagram**

**FREQUENCY PLANNING** – The process for FCC authorization and IARU frequency coordination of amateur radio satellite frequencies is well known and regularly practiced by AMSAT-NA. All Fox-1x satellite frequency plans have been (will be) determined iaw current FCC/IARU policies and direction. These include adherence to the IARU's Amateur Satellite Service coordinated frequency allocations, specifically uplinks and downlinks in the 144 MHz to 146 MHz (145.8 MHz to 146.0 MHz preferred)[1] and 435MHz to 438MHz.

AMSAT-NA Vice President of Operations, Drew Glasbrenner (KO4MA) has proposed that Fox-1E's linear transponder seek coordination of a 30 kHz transponder frequency allocation, specifically an **uplink frequency of 145.860 MHz to 145.890 MHz** and **downlink frequency of 435.760 MHz to 435.790 MHz**. This allocation would result in a third harmonic (transmit ↔ receive) separation of approximately 1.8 to 1.9 MHz for amateur satellite ground stations. These frequencies will constitute AMSAT's Fox-1E Amateur Satellite Service allocation coordination request to the IARU/FCC and will be used as the receive and transmit frequencies for this signal chain analysis and spurious emissions analysis.

Knowing the transmit/receive frequencies, a frequency plan must then reflect the choice of an appropriate down and up conversion scheme for the transponder's mixers. The Fox-1E linear transponder's mixers, like all mixers, approximately follow an exponential current-voltage characteristic transfer function. This nonlinear operation results in the production of harmonics at all integral frequencies of the fundamental Radio Frequency (RF) and Local Oscillator (LO) tones, and intermodulation frequencies at all integral combinations of the form:

$\pm m \bullet RF \pm n \bullet LO$ where: m and n are integers,
RF is the signal frequency, and
LO is the local oscillator frequency.

Effective RF design requires judicious selection of LO frequencies and filtering of RF signals to eliminate or minimize undesired extra, or spurious frequencies generated by the mixers. Generally, frequencies generated by a mixer can be filtered at the Intermediate Frequency (IF). However, RF designers must consider incoming signals that result in mixer output products that fall within the

bandwidth of the IF. These result in false/aliased IF signals which can mask or interfere with the desired input signal(s).

[1] In accordance with the ARRL Band Plan. That plan also includes a "new OSCAR sub-band" of 144.3 MHz to 144.5 MHz. However, current North American APRS operations on 144.39 MHz as well as other established weak signal communications are active in that spectrum.

Normally an RF designer concerned with efficient frequency planning is only constrained by the requirement to receive (and transmit) within a particular frequency band/spectrum. However, for Fox-1E, the LO frequency is further constrained by AMSAT's "prior purchase/stock" of 45 MHz crystal filters. Therefore, given that the RF and IF frequencies have been so determined, the only frequency planning variable remaining is to decide on high or low-side injection for the LO frequencies. In a receiver, when the LO frequency falls below the desired RF, it is called *low-side injection* and the mixer a low-side down-converter; when the LO falls above the desired RF, it is called *high-side injection* and the mixer a high-side down-converter.

In order to reduce the (frequency shifting) effects of Doppler, it is preferable to design an INVERTING transponder. Inverting frequency operation is the result of spectral inversion of the RF signals. Spectral inversion is achieved through high-side frequency injection in the system's mixers. No spectral inversion occurs with low-side injection. Therefore to achieve an inverting transponder design, implementing the same side injection in the transponder's RX (down-conversion) and TX (up-conversion) mixers would NOT achieve any spectral inversion, i.e., inverting operation. The design decision then becomes which mixer should be low-side injected and which should be high- side injected. So the question could be asked - is the RX (receive) down-conversion mixer implemented with high- or low-side injection? The answer to that question then determines the injection side of the (transmit) up-conversion mixer. Given the receive (down-conversion) mixer frequency plan has been established by the input frequencies (145.860 MHz to 145.890 MHz) and a 'constrained' IF (45 MHz), then the two possible RX down (block) conversion mixer injection fixed frequencies can be calculated as:

$$[(145.890 + 145.860) \div 2] \pm 45 = 100.875 \text{ MHz or } 190.875 \text{ MHz for low/high-side injection}$$

The choice between these two frequency candidates is influenced by the desire to generate a stable, low drift LO which is easy to design and realize. This could motivate an RF designer to choose low-side injection. However, considering potential interference from any harmonics generated by both low and high LOs, the 4$^{th}$ harmonic of $LO_{low}$ is 32.26 MHz away from the TX frequency band low edge of 435.760 MHz, while the 2$^{nd}$ harmonic of $LO_{high}$ is 54.01 MHz away. Furthermore, a high-side RX injected down converter would afford a lower frequency (low-side injected) TX up converter LO frequency below 435 MHz. Consequently, considering these factors, the receive mixer LO design was chosen to employ high-side injection.

The input RF band and IF have already been specified, so RX high-side injection dictates an IF center frequency of:

$$\mathbf{IF = LO_{RX} - RF_{2m} \text{ or } LO_{RX} = RF_{2m} + IF = 145.875 \text{ MHz} + 45.000 \text{ MHz} = 190.875 \text{ MHz}}$$

Since the Fox-1E linear transponder's transmit mixer up-converts a 45 MHz signal to 70cm, this can be accomplished as RF = IF + LO. The LO frequency required would then be:

$$\mathbf{RF_{70cm} = IF_{45} + LO_{TX} \text{ or } LO_{TX} = RF_{70cm} - IF_{45} = 435.775 \text{ MHz} - 45 \text{ MHz} = 390.775 \text{ MHz}}$$

This operation represents low-side injection and therefore complements the RX high-side injection resulting in spectrum inversion.

The Fox-1E's linear transponder RX high-side and TX low-side injection scheme is therefore shown in Figure 2 below:

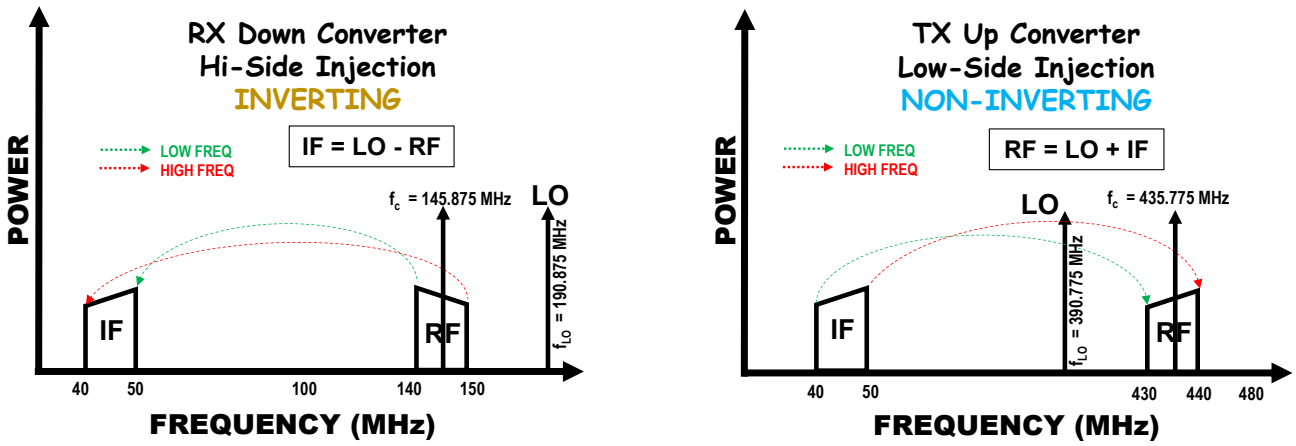



**Figure 2: Fox-1E Linear Transponder UP-DOWN Conversion Scheme**

**Receive Frequency Planning** - The spectral implications of high-side RX LO injection using the frequencies derived above can be illustrated using graphical software. The result is a Frequency Plan Chart that shows the RF band, IF band, and LO frequency. Specifically, the RF and IF bands and LO are graphed and any spurious products resulting are plotted referenced to the input RF frequency. Such a plot for the Fox-1E linear transponder receive frequencies is shown in Figure 3. In-band spurs are shown in **red**; near-band spurs are shown in **purple**, and the legend and text output detail the ±m • RF x ±n • LO spur order and level.

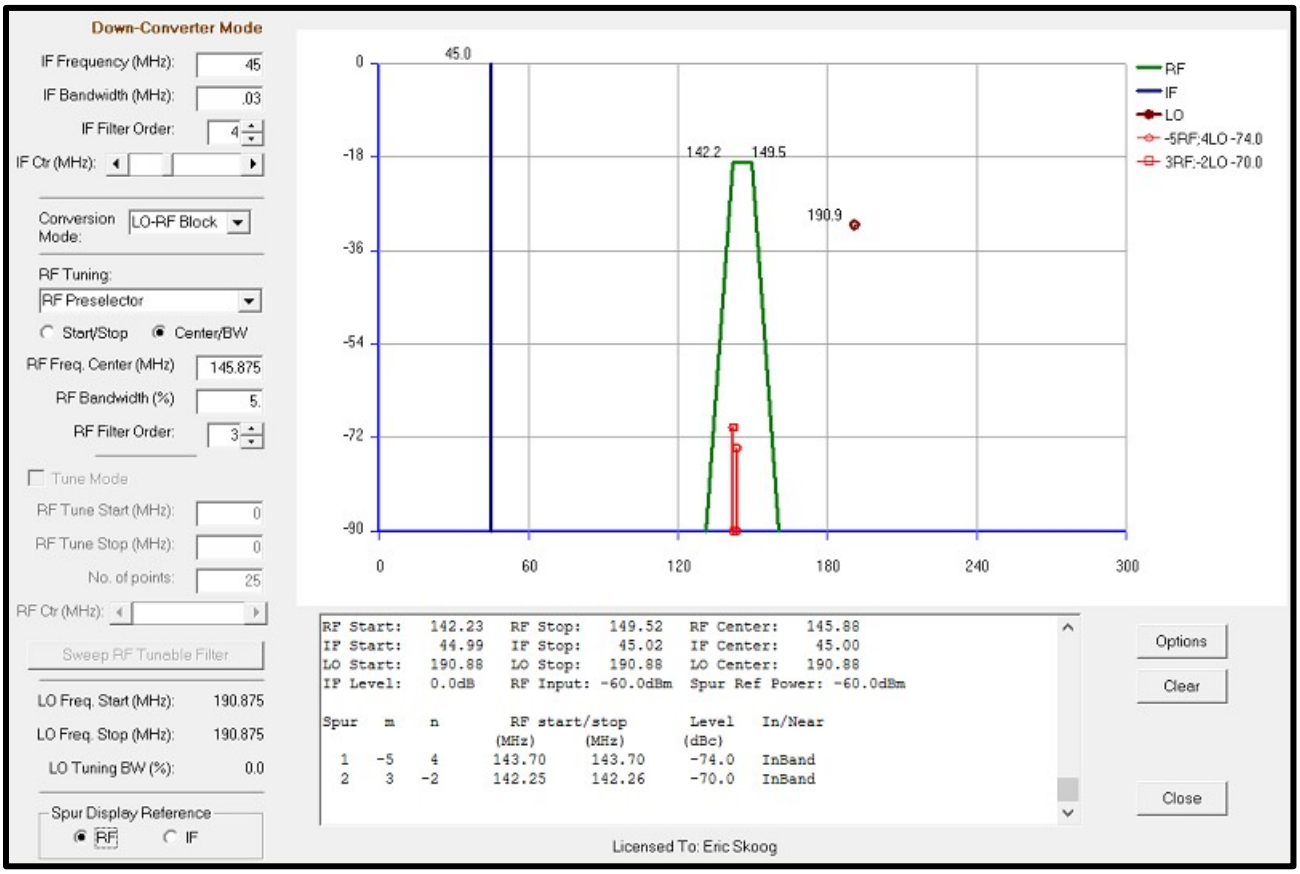



**Figure 3: Fox-1E Linear Transponder Receive Frequency Plan Chart**

The frequency (In-)band range of interest (142.23 MHz to 149.52 MHz) shown in Figure 3 was chosen to include the receive range of uplink Command Channel frequencies **(145.8 to 145.860 MHz and 145.900 MHz to 146 MHz)** that are out-of-(transponder) band but still within the ARRL's satellite band plan. Including this extended range, the resultant spur picture shows there are no transponder in-band spur products and only two (143.7 MHz and 142.255 MHz) spurs outside the satellite band plan frequencies, and those are down more than -70 dBc!

Another view of the spurious products resulting from the frequency plan is a ("Spider Web") Spur Chart. A Spur Chart plots the frequencies at which spurs occur between the input and output frequencies according to the mixer mode implemented. Clear areas along the desired output frequency range are spurious-free areas that indicate frequency candidates for a down-converting IF operating frequency. In such a chart, the X and Y axis can be set as the IF, RF, or LO for maximum flexibility in finding spur-free zones.

The plot is made over a larger frequency range than the desired operating frequency range, yielding a broader view of where and how spurious frequencies are being generated. This is especially important for the Fox-1E linear transponder due to the inclusion of an out-of-(transponder) band Command and Control (receive) channel. The **green** box displays the input and output operating frequency range to allow identification of any spurs falling within that range.

One useful form of the Spur Chart shows the output IF frequency as a function of the input RF frequency to the mixer. This is the case in Figure 4. In this chart the lines represent specific spurs the mixer generates and all other spurious products are color coded according to their relative levels. The mixer product range is in **light blue**.

The **green** IF ↔ RF range box shown is from 44.5 MHz to 45.5 MHz (100 kHz) and the amateur radio 2 meter RF band frequency range (145.4 MHz to 146.5 MHz). The graph shows no spurs present in that range.

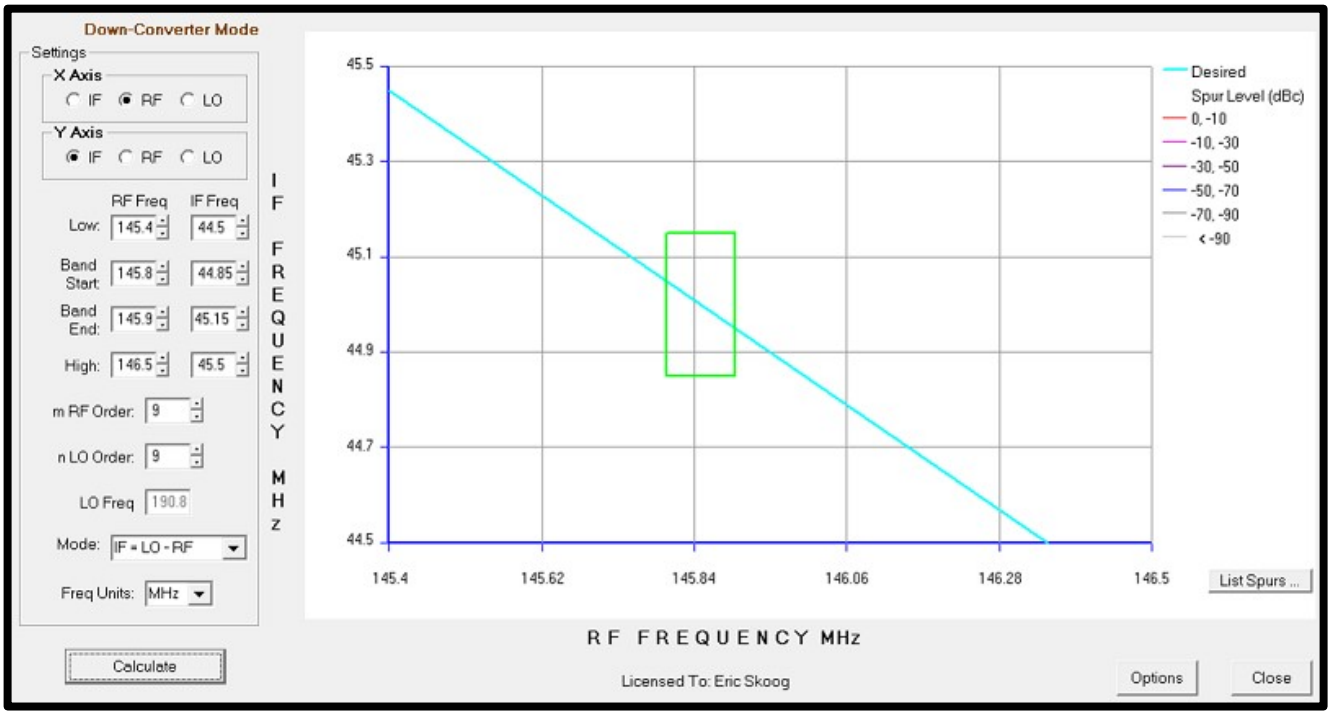



**Figure 4: Fox-1E Linear Transponder Receive Spur Chart**

**Transmit Frequency Planning** – Transmit signal chain frequency planning involves low-side injection with an LO frequency of 390.775 MHz for TX up-conversion as calculated earlier. This transmit frequency plan is shown in the Frequency Plan Chart of Figure 5. The up-converted IF is plotted with a bandwidth of 1% (44.78 MHz to 45.23 MHz) which includes a 44.965 MHz IF-injected 1200 bps BPSK telemetry beacon downlink channel up-converted to an out-of-(transponder)-band **435.750 MHz**. As this transmit frequency planning figure shows, the transmit spectrum contains no spurious products.

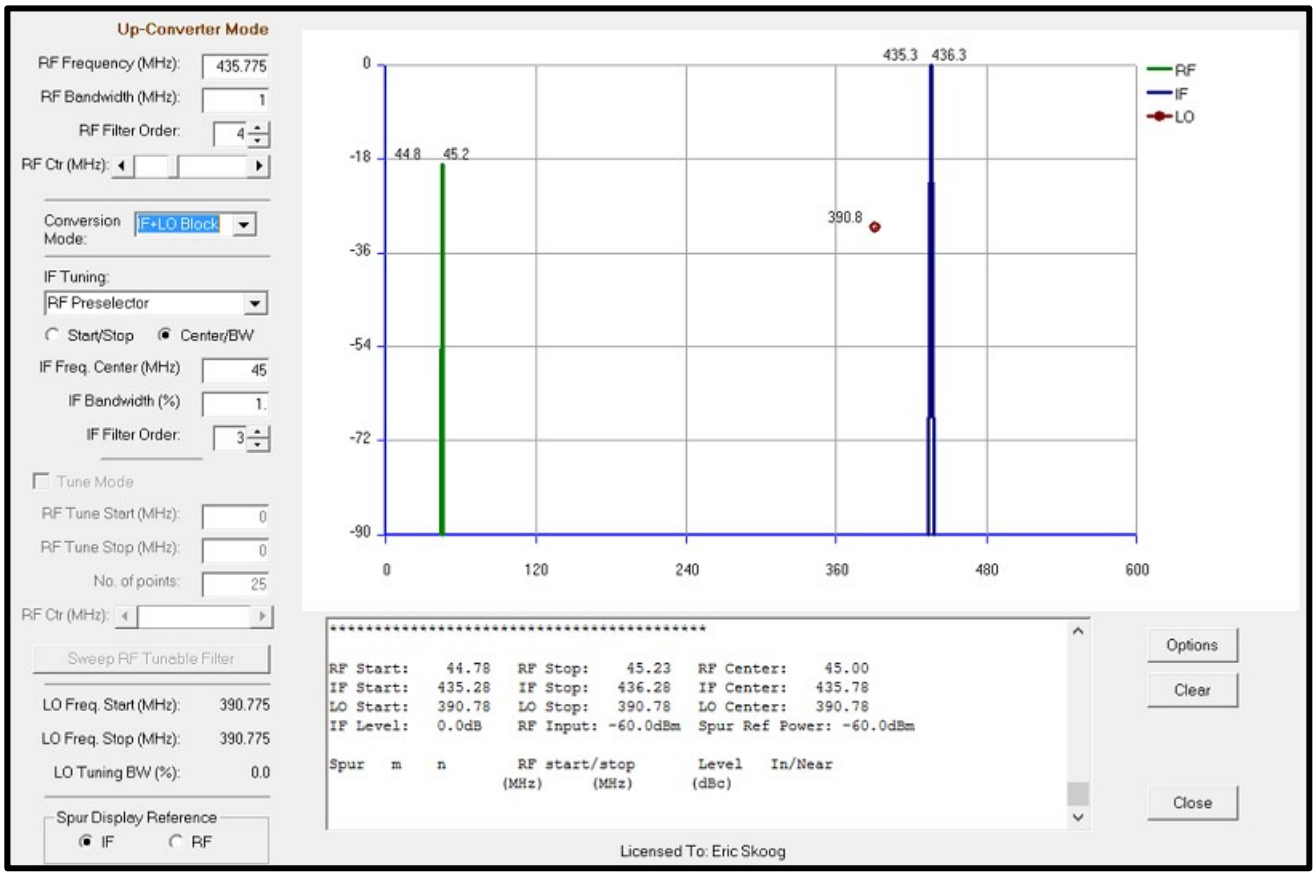



**Figure 5: Fox-1E Linear Transponder Transmit Frequency Plan Chart**

Likewise, the Spur (Web) Chart for the transmit signal chain is shown in Figure 6. This chart's green box shows the up-conversion process implemented using the frequencies chosen results in no spurs in the transmit frequency range of interest (435 MHz to 438 MHz). That range includes the 1200 BPSK downlink telemetry channel at 435.750 MHz.

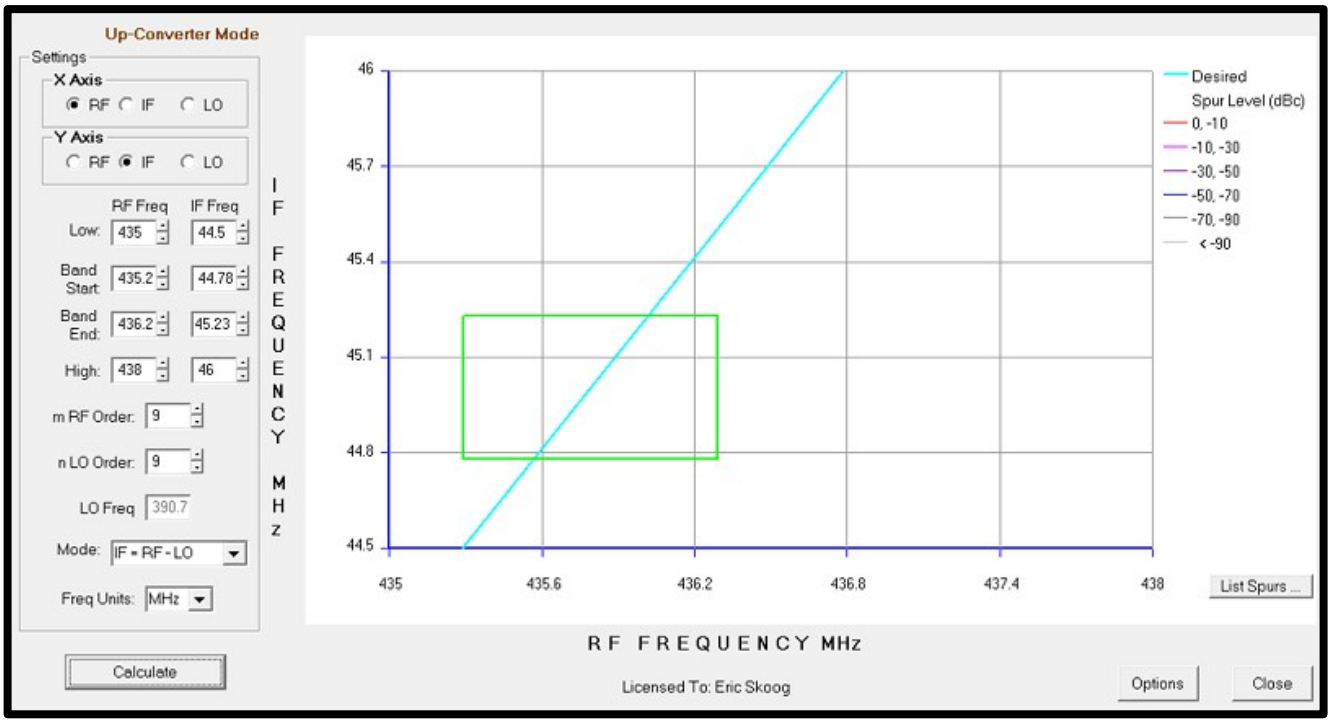



**Figure 6: Fox-1E Linear Transponder Transmit Spur Chart**

**DESIREMENTS** – Having determined a frequency plan, attention can be directed to considering the extent to which the proposed, initial design is capable of satisfying the users *unconstrained wishes*. For RF systems this means articulating performance via a set of "desirements." These are merely (performance) requirements that would be ideal IFF they could be satisfied by an affordable, viable system design. Thus, although the RF design may start out with a wished-for list of 'desirements,' as the system's analyses, design, trade-offs, test and refinement progress, such a desirements list is necessarily relaxed to a set of firm requirements.

When studying an initial desirements list for "gold-plating," it is helpful to consider 'corner cases' (use-cases) to ensure the system under design will be beneficial to the operators. One initial use-case that will assist scoping the low end of the communications analysis is the 'disadvantaged' ground station; disadvantaged in the sense that an operator's transmit/receive capabilities are less than a fixed, normally equipped satellite ground station. For example, consider a low-power, portable amateur satellite service station regularly used by operators consisting of a Yaesu FT-817 portable multi-mode transceiver running 5 watts SSB transmit power into an Arrow II 2 m/70 cm antenna. This use-case establishes the desired low-end performance required of the system. Given such a station as the low-end, boundary case, a set of operational link budgets can be calculated as follows:

**Transponder Uplink Signal Calculation** – Using standard AMSAT – IARU link budget calculation spreadsheets authored by Jan King (W3GEY/VK4GEY) et al, the expected uplink signal performance can be calculated as shown in Table 1. The "goes-into" data entries that underpin these calculations are depicted in **Blue** text. They originate from the 'disadvantaged' (corner case) chosen equipment performance parameters detailed above, link budget equation constants/formulae, expected orbital parameters, and a few estimated/assumed tracking condition numbers. The orbital parameters have been chosen to be typical of a low altitude LEO trajectory. The satellite's slant range is calculated at the worst case (max. range) obtained from an elevation angle of 1°. Other inputs consist of the linear transponder's receive bandwidth of 30 kHz, the mid-frequency of the transponder's receive bandwidth, $f_c$ = 145.885 MHz; and assumed transponder noise figure, losses, and gains. **Red** and **Black** entries represent calculated values from the equations shown in the comments column of the spreadsheet.

As Table 1 shows, with the inputs and assumptions stated, the worst case uplink margin calculated is approximately **19.6 dB** when calculated in the 30 kHz Linear Transponder bandwidth or 10 dB better, **29.6 dB** in a nominal 3 kHz user bandwidth. Thus for a low-end, 'disadvantaged' ground station transmitting 5 watts through a hand-held antenna, the probability of closing the uplink is high. Note that this worst case calculation is predicted on maximum slant range, i.e., 2,398 km at 1° elevation, path loss. This worst case slant range distance reduces to a "BEST CASE" 475 km at zenith which equates to a reduction in path loss of approximately 14 dB.

**Transponder Downlink Signal Calculation** – The Fox-1E linear transponder's UHF downlink budget, assuming an average envelope power (shared) of 27 dBm (500 mW) transmitting in the 70 cm Amateur Satellite Service allocation with a center frequency $f_c$ = 435.785 MHz is shown in Table 2. Again, like the uplink, these calculations assume a worst case slant range distance for a 1° antenna elevation angle and typical, portable Yaesu FT-817 transceiver reception. Additionally, the communication downlink channel is assumed to be carrying five (5) user signals, of 3 kHz each (including the TLM/EXP dedicated downlink channel). As Table 2 shows, these user and orbit conditions result in a downlink margin of **9.14 dB**. Again, this figure could improve for a (BEST CASE) overhead (zenith) elevation by approximately 14 dB. Therefore, given a low end, disadvantaged, portable station, it appears highly probable that the UHF downlink can be successfully closed.

**Fox-1E CubeSat**

# V/U Analog Linear Transponder

## Uplink Transponder Budget:

NOTE:

Eric Skoog (K1TVV)

Version: 1.6.1

Date Data Last Modified:

21-Oct-16

Uplink Center Frequency: 145.885 MHz

| Parameter: | Value: | Units: | Comments: |
|---|---|---|---|
| *Ground Station:* | | | |
| User Uplink Transmitter Power Output ($P_{G,out}$): | 5.0 | watts | This is the power associated with ONE uplinking user station. |
| in dBW: | 7.0 | dBW | |
| in dBm: | 37.0 | dBm | |
| Transmission Line Losses ($L_{G,TL}$): | -1.0 | dB | |
| Connector, Filter or In-Line Switch Losses ($L_{G,FIL}$): | -1.0 | dB | |
| Antenna Gain ($G_{G,A}$): | 7.0 | dBi | |
| Ground Station Effective Isotropic Radiated Power ($EIRP_{G,T}$): | 12.0 | dBW | Ground Station Effective Isotropic Radiated Power: $EIRP_{G,T} = P_{G,out} + L_{G,TL} + L_{G,FIL} + G_{G,A}$ |
| Ground Station Antenna Pointing Loss ($L_{G,PT}$): | -1.0 | dB | |
| *Uplink Path:* | | | |
| Antenna Polarization Losses ($L_{POL}$): | -2.50 | dB | |
| Path Loss ($L_{PATH}$): | -143.3 | dB | $L_{PATH} = 20Log(4\pi) + 20Log(R_{slant}/\lambda)$ |
| Atmospheric Losses ($L_{ATMOS}$): | -4.6 | dB | Use value appropriate for elevation angle selected in Orbit Performance W/S; Reference: see Ippolito's book. |
| Ionospheric Losses ($L_{ION}$): | -0.4 | dB | Reference Ippolito again. |
| Rain Losses ($L_{RAIN}$): | 0.0 | dB | Negligable at this uplink frequency |
| Isotropic Signal Level at the Spacecraft ($EIRP_{S,R}$): | -139.9 | dBW | $EIRP_{S,R} = EIRP_{G,T} + L_{G,PT} + L_{POL} + L_{PATH} + L_{ATMOS} + L_{ION} + L_{RAIN}$ |
| *Spacecraft:* | | | |
| Spacecraft Rcvr Antenna Pointing Loss ($L_{S,PT}$): | -0.1 | dB | |
| Spacecraft Rcvr Antenna Gain ($G_{S,A}$): | 2.2 | dBi | |
| Spacecraft Transmission Line Losses ($L_{S,TL}$): | -0.1 | dB | |
| Spacecraft Input Filter Insertion Loss ($L_{S,FIL}$): | -1.0 | dB | |
| Spacecraft LNA Noise Temperature ($T_{S,LNA}$): | 60 | K | |
| Spacecraft Transmission Line Temp. ($T_{S,TL}$): | 273 | K | |
| Spacecraft Sky Temperature ($T_{S,SKY}$): | 290 | K | S/C receive antenna sees earth temperature. |
| S/C Transmission Line Coefficient ($\alpha$): | 0.7762 | | $\alpha = 10^{\wedge}\{(L_{S,TL} + L_{S,FL})/10\}$ |
| Spacecraft Effective Noise Temperature ($T_{S,Neff}$): | 346 | K | $T_{S,Neff} = \alpha(T_{S,SKY}) + (1 - \alpha)T_{S,TL} + T_{S,LNA}$ |
| Spacecraft Figure of Merrit ($G/T_S$): | -24.3 | dB/K | $G/T_S = G_{S,A} + L_{S,TL} + L_{S,FIL} - 10LogT_{S,Neff}$ |
| **S/C Signal-to-Noise Power Density ($S/N_{o,S}$):** | 64.4 | dBHz | Boltzmann's Constant: k = -228.6 dBW/K Hz  $S/N_{o,S} = EIRP_{S,R} + L_{S,PT} - k + G/T_S$ |
| Transponder IF Bandwidth ($B_{T,IF}$): | 30.0 | kHz | |
| Transponder Uplink Input Noise Power ($N_{T,IN}$) | -158.4 | dBW | $N_{T,IN} = kTB = k + 10LogT_{S,Neff} + 10Log(B_{T,IF} \times 10^3)$; i.e., Additive White Gaussian Noise (AWGN); |
| **Single User Uplink S/N in Transponder Bandwidth ($S/N|_{SU,T}$):** | 19.6 | dB | This is the S/N for ONE user seen at the S/C Rcvr IF, measured after the BPF, in the bandwidth determined by that filter. |
| | | | $S/N|_{SU,T} = S/N_{o,S} - 10Log(B_{T,IF} \times 10^3)$ |
| Transponder Uplink Interference Density ($I_{O,T}$): | -250.0 | dBW/Hz | This is the interference density seen at the S/C Receiver. Assume Uniform Density Across Transponder Passband. |
| Transponder Uplink Total Interference Power ($I_{T,total}$): | -203.2 | dBW | Measured at Spacecraft Receiver Input: $I_{T,total} = I_{O,T} + 10Log(B_{T,IF} \times 10^3) + L_{S,PT} + G_{S,A} + L_{S,TL}$ |
| Transponder Uplink S/I Power Ratio: | 63.4 | dB | This is the ratio of the user signal level compared to the Interference Noise in the entire Transponder Passband; $S/I = EIRP_{S,R} - I_{T,total}$ |
| Transponder Total Uplink Noise Power ($N_{T,N+I}$): | -158.4 | dBW | This total noise multiplied by the transponder's gain will produce wasted output noise power from the HPA; $N_{T,N+I} = 10Log[10^{\wedge}(N_{T,IN}/10) + 10^{\wedge}(I_{T,total}/10)]$ |
| Single User $S/(N+I)|_{SU,T}$ in Transponder Bandwidth: | 19.6 | dB | This is the uplink performance measured in the ENTIRE transponder bandwidth (NOTE: This could be a negative number) |
| | | | $S/(N+I)|_{SU,T} = 10Log\{1/[(1/(10^{(S/N|_{SU,T})/10})) + (1/10^{(S/I)/10})]\}$ |
| Single User Signal Bandwidth ($B_{T,SU}$): | 3.0 | kHz | |
| | | | |
| **Single User Uplink $S/(N+I)|_{SU,t}$ in User Terminal Bandwidth:** | 29.58 | dB | THE BOTTOM LINE FOR THE UPLINK (NOTE: This is the average S/(N+I), not the peak value. |
| | | | $S/(N+I)|_{SU,t} = S/(N+I)|_{SU,T} + 10Log(B_{T,IF}/B_{T,SU})$ |

**Table 1: Fox-1E Linear Transponder VHF Uplink RX Signal Link Budget Calculations**

**Fox-1E CubeSat**

## V/U Analog Linear Transponder

## Downlink Transponder Budget:

NOTE:

Eric Skoog (K1TVV) Date Data Last Modified:

Version: 1.6.1 21-Oct-16

**DownLink Center Frequency: 435.785 MHz**

| Parameter: | Value: | Units: | Comments: |
|---|---|---|---|
| *Spacecraft:* | | | |
| Spacecraft True HPA Power Output per User Channel ($P_{S,out,true}$) | 0.0979 | watts | This is the "True" RF Transponder Power Allocated to Each User. The Intermod Power is **NOT** Included in this value. |
| In dBW: | -10.09 | dBW | $P_{S,out,true} = \{P_{out,total} / \#\ of\ Channels\} / \{(1 + 10^{(-S/N+I|_{SU,T}/10)} + 10^{(-IMR)}) + [10^{(-S/N+I|_{SU,T}/10)}][10^{(-IMR)}]\}$ |
| In dBm: | 19.91 | dBm | |
| Spacecraft HPA Transponded Uplink Noise Power: ($N_{S,TRD}$) | 0.0011 | watts | This is the uplink noise power amplified and retransmitted by the transponder HPA - Per Channel |
| | | | $N_{S,TRD} = P_{S,out,true} \times 10^{(-S/N+I|_{SU,T})}$ |
| Spacecraft Transponded Intermod Power ($P_{S,IMD}$): | 0.0010 | watts | This is the HPA power wasted in generating intermodulation products - Per Channel |
| | | | $P_{S,IMD} = (P_{S,out,true}) \times (10^{(-S/N+I|_{SU,T}/10)}) \times (10^{(-IMR/10)}) + (P_{S,out,true}) \times (10^{(-IMR/10)})$ |
| Spacecraft Total HPA Power Allocated per User Channel ($P_{S,out,total}$): | 0.1000 | watts | $P_{S,out,total} = P_{S,out,true} + N_{S,TRD} + P_{S,IMD}$ - Per Channel |
| Spacecraft Transmission Line Losses ($L_{S,TL}$): | -0.1 | dB | |
| S/C Connector, Filter and In-Line Switch Losses ($L_{S,FIL}$): | -1.0 | dB | |
| Spacecraft Transmit Antenna Gain ($G_{S,A}$): | 2.3 | dBi | |
| Spacecraft EIRP per User ($EIRP_{S,SU}$): | -8.89 | dBW | Spacecraft Effective Isotropic Radiated Power: $EIRP_{S,SU} = P_{S,out,true} + L_{S,TL} + L_{S,FIL} + G_{S,A}$ per User |
| Spacecraft Transmit Antenna Pointing Loss ($L_{S,PT}$): | -0.1 | dB | |
| *Downlink Path:* | | | |
| Antenna Polarization Loss ($L_{S,POL}$): | -2.5 | dB | |
| Path Loss ($L_{PATH}$): | -152.8 | dB | |
| Atmospheric Loss ($L_{ATMOS}$): | -4.6 | dB | Use Value Appropriate for Elevation Angle Selected in Orbit Performance W/S. See Ippolito. |
| Ionospheric Loss ($L_{ION}$): | -0.4 | dB | Reference Ippolito |
| Rain Loss ($L_{RAIN}$): | 0.0 | dB | |
| Isotropic Signal Level at Ground Station ($EIRP_{G,R}$): | -169.3 | dBW | -139.3 dBm $EIRP_{G,R} = EIRP_{S,SU} + L_{S,PT} + L_{S,POL} + L_{PATH} + L_{ATMOS} + L_{ION} + L_{RAIN}$ |
| *Ground Station:* | | | |
| Ground Station Antenna Pointing Loss ($L_{G,PT}$): | -1.0 | dB | |
| Ground Station Antenna Gain ($G_{G,A}$): | 12 | dBi | |
| Ground Station Transmission Line Losses ($L_{G,TL}$): | -1.0 | dB | |
| Ground Station Filter and/or Switch Losses ($L_{G,FIL}$): | -1.0 | dB | |
| Ground Station LNA Noise Temperature ($T_{G,LNA}$): | 80 | K | |
| Ground Station Transmission Line Temp. ($T_{G,TL}$): | 290 | K | |
| Ground Station Sky Temperature ($T_{G,SKY}$): | 100 | K | Grd Station Antenna looks at cold sky. |
| Ground Station Transmission Line Coefficient ($\alpha$): | 0.6310 | | $\alpha = 10^{\{(L_{G,TL} + L_{G,FIL})/10}$ |
| Ground Station Effective Noise Temperature ($T_{G,Neff}$): | 250 | K | $T_{G,Neff} = \alpha(T_{G,SKY}) + (1 - \alpha)T_{G,TL} + T_{G,LNA}$ |
| Ground Station Figure of Merrit ($G/T_G$): | -14.0 | dB/K | $G/T_G = G_{G,A} + L_{G,TL} + L_{G,FIL} - 10LogT_{G,Neff}$ |
| Ground Station Signal-to-Noise Power Density ($S/N_{o,G}$): | 44.28 | dBHz | Boltzman's Constant: k = -228.6 dBW/K/Hz |
| | | | $S/N_{o,G} = EIRP_{G,R} + L_{G,PT} - k + G/T_G$ |
| Transponder Intermodulation Ratio (S/IM or IMR): | 20.0 | dB | The ratio of the power in one user signal divided by the average intermodulation power level in one User channel |
| Ground Station Signal-to-Intermodulation Power Density ($S/I_o$): | 54.8 | dBHz | $S/I_O = EIRP_{G,R} - (EIRP_{G,R} - IMR - 10LogB_{T,SU}) + 30$ |
| Ground Station $S/(N_o+I_o)$: | 43.91 | dBHz | $S/(N_O+I_O) = 10Log\{1/[(1/(10^{((S/N_{su,T})/10)}) + (1/(10^{((S/I)/10)})]\}$ |
| Single User Signal Bandwidth ($B_{G,SU}$): | 3.0 | kHz | |
| **Single User Downlink S/N in User Terminal Bandwidth:** | 9.14 | dB | This is the S/N for ONE user due to downlink thermal noise (AWGN) plus Intermodulation Interference. |
| | | | It does not include the effects of the uplink noise. |
| | | | $S/N|_{SU,t} = S/(N_O+I_O) - 10Log(B_{G,SU} \times 1000)$ |

**Table 2: Fox-1E Linear Transponder Downlink TX Signal Link Budget Calculations**

**Combined Uplink - Downlink Calculation** – The last step in link budgeting for a "bent-pipe" linear transponder is to combine the up- and down-link calculations to determine the system's end-to-end performance. Table 3 presents a summary of the overall system under the conditions of LOS propagation in clear sky for a **WORST CASE** elevation angle of 1°. Included in this analysis is an allocation for interference assumed to be from sources such as cross-polarization, adjacent satellite encroachment, earth-originating EMI, etc. This type of entry is necessary because all operating satellites are exposed to sources of interference. The uplink-downlink signal budget calculations show that the Fox-1E Linear Transponder's communication links can be closed.

**Transponder Link Results (Combining the Uplink and the Downlink):** Version: 1.6.1

*Link Results:*

| | |
|---|---|
| Uplink S/N Achieved: | 29.58 dB |
| Downlink S/N Achieved: | 9.14 dB |
| **Combined Uplink + Downlink S/N Achieved:** | **9.10 dB** |
| | *If Applicable* |
| SSB PEP S/N: | 19.10 dB |

*Spacecraft Resources:*

| | |
|---|---|
| Spectrum Used: | 30 kHz |
| Users & Channels Supported: | 5 User Ch.<br>3.0 kHz/Ch. |
| Transponder End-to-End Gain: | 134.8 dB |
| Transponder DC Power Consumed: | 0.83 Watts |
| Transponder Thermal Dissipation: | 0.33 Watts |

*User Uplink:*

| | |
|---|---|
| Uplink Center Frequency: | 145.885 MHz |
| Uplink: User Antenna Gain: | 7.0 dBi |
| Uplink: User Transmitter Power: | 5.0 Watts |

*User Downlink:*

| | |
|---|---|
| Downlink Center Frequency: | 435.785 MHz |
| Downlink: User Antenna Gain: | 12 dBi |
| Downlink: User LNA Temperature: | 80 K |
| Downlink: User Rcvr G/T: | -13.98 dB/K |

*Spacecraft:*

| | |
|---|---|
| S/C Uplink Antenna Gain: | 2.2 dBi |
| S/C Uplink G/T: | -24.29 dB/K |
| S/C Downlink Total Xmtr Power: | 0.50 Watts |
| S/C Downlink RF Power/User Ch.: | 0.10 Watts |
| S/C Downlink Antenna Gain: | 2.3 dBi |
| S/C Downlink EIRP (Transponder): | -1.8103 dBW |
| S/C Downlink EIRP/User Ch.: | -8.9 dBW |

**Table 3: Fox-1E Linear Transponder RX/TX Combined Signal Link Budget Summary**

**TLM/EXP Downlink Signal Calculation** – The Fox-1E linear transponder design includes a (transponder) out-of-band, dedicated 1200 bps BPSK-modulated channel to downlink beacon telemetry and experiment data. The performance of this channel is critical to monitoring the health and welfare of the satellite and transmitting experiment data to ground-based analysis centers. Consequently, it is imperative that the probability of adequate signal reception be maximized to support data decoding and information retrieval. Given the operational constraints of limited downlink power and bandwidth, the trade-off to consider is redundancy vs. throughput by employing Forward Error Correction coding (FEC). There are many types of FEC codes that could be used. It is well known that of the major FEC types, convolutional codes work well for environments characterized by weak signals and Additive White Gaussian Noise (AWGN) caused random errors, while Linear Block Codes perform well under burst error conditions. Considering the operational environment of the Fox-1E CubeSat's Low Earth Orbit, link signal margins are above weak signal levels which should prove sufficient to close the downlink. However, satellite rotation, antenna pattern discontinuities, and changing slant range geometry for ground stations will result in slow fading (burst error) conditions. Consequently, it was decided to continue to employ Block Coding in the Fox-1E satellite design as had been used in AMSAT's previous Fox-1A through 1D CubeSats. Therefore, the TLM/EXP downlink performance was calculated using the standard Reed Solomon

(255, 223) block code. Beyond that, most of the downlink parameters used in the calculation are the same as used in the Linear Transponder communications downlink. However, as noted earlier, the TLM/EXP downlink frequency lies outside of the Linear Transponder's 30 kHz passband, but shares the same output power available to the passband users. Consequently, it is assumed that the 'nominal' power output of the TLM/EXP channel will be approximately 100 mW (20 dBm). Lastly, it is assumed that the majority of telemetry and experiment data collectors will be using standard ground stations. Therefore, the link budget calculations take advantage of relatively high gain, directional

| **Fox-1E CubeSat** — **TLM/EXP Downlink Channel Link Budget** | | | **AMSAT/IARU Annotated Link Model System** Version: 2.5.3.1 from Jan A. King (W3GEY/VK4GEY) | Date Data Last Modified: 10/22/2016 |
|---|---|---|---|---|
| *Parameter:* | *Value:* | *Units:* | *Comments:* | |
| ***Spacecraft:*** | | | | |
| Spacecraft Transmitter Power Output ($P_{S,out}$): | 0.10 | watts | This value is transferred from "Transmitters" W/S, Cell [E50] | |
| In dBW: | -10.0 | dBW | Transmitter power expressed in dB above one watt | |
| In dBm: | 20.0 | dBm | Transmitter power expressed in dB above one milliwatt | |
| Spacecraft Total Transmission Line Losses ($L_{S,TLtotal}$): | 1.3 | dB | This value is transferred from "Transmitters" W/S, Cell [I71] | |
| Spacecraft Antenna Gain ($G_{S,A}$): | 2.3 | dBi | | |
| Spacecraft EIRP ($EIRP_{S,T}$): | -9.0 | dBW | Spacecraft Effective Isotropic Radiated Power: $EIRP_{S,T} = P_{S,out} - L_{S,TLtotal} + G_{S,A}$ | |
| ***Downlink Path:*** | | | | |
| Spacecraft Antenna Pointing Loss ($L_{S,PT}$): | 0.1 | dB | This value is calculated in the "Antenna Pointing Losses" W/S, and trasferred from Cell [K85] | |
| Spacecraft-to-Ground Antenna Polarization Loss ($L_{POL}$): | 0.2 | dB | This value is calculated in the "Polarization Loss" W/S and is transferred from Cell [F60]. | |
| Path Loss ($L_{PATH}$): | 152.8 | dB | $L_{PATH} = 22 + 20LOG(D/\lambda)$; Transferred from "Frequency" W/S, Cell [M18] | |
| Atmospheric Loss ($L_{ATMOS}$): | 4.6 | dB | This value is transferred from "Atmos. & Ionos. Losses" W/S, Cell [D23] (for longest slant range) | |
| Ionospheric Loss ($L_{ION}$): | 0.8 | dB | This value is transferred from "Atmos. & Ionos. Losses" W/S, Cell [D45:D48] | |
| Rain Loss ($L_{RAIN}$): | 0.0 | dB | This value should be estimated by the link model operator and placed into Cell [B18] | |
| Isotropic Signal Level at Ground Station ($EIRP_{G,R}$): | -167.6 | dBW | This is the signal level received at the Earth in the vacinity of the ground station using an omnidirectional antenna. $EIRP_{G,R} = EIRP_{S,T} - (L_{S,PT} + L_{POL} + L_{PATH} + L_{ATMOS} + L_{ION} + L_{RAIN})$ | |
| ***Ground Station (EbNo Method):*** ***-------- Eb/No Method --------*** | | | | |
| Ground Station Antenna Pointing Loss ($L_{G,PT}$): | 0.1 | dB | This value is transferred from "Antenna Pointing Losses" W/S, Cell [K102] | |
| Ground Station Antenna Gain ($G_{G,A}$): | 15.5 | dBi | This value is selected at "Antenna Gain" W/S, Cell [E58] (assuming a yagi antenna) | |
| Ground Station Total Transmission Line Losses ($L_{G,TLtotal}$): | 0.2 | dB | This value is transferred from the "Receivers" W/S, Cell [J124] | |
| Ground Station Effective Noise Temperature ($T_{G,Neff}$): | 1597 | K | This value is calculated in the "Receivers" W/S and Transferred from Cell [J147] | |
| Ground Station Figure of Merrit ($G/T_G$): | -16.7 | dB/K | $G/T_G = G_{G,A} - L_{G,TLtotal} - 10log(T_{G,Neff})$. This is the ultimate measure of the receiver's performance. | |
| Ground Station Signal-to-Noise Power Density ($S/N_{oG}$): | 44.2 | dBHz | Boltzman's Constant: k = -228.6 dBW/K/Hz $S/N_{oG} = EIRP_{G,R} - L_{G,PT} - k + G/T_G$ | |
| System Desired Data Rate ($R_{bit}$): | 1200 | bps | Operator selects this value. Be Careful! This is the data rate, not the symbol rate. | |
| In dBHz: | 30.8 | dBHz | This is simply = $10log(R_{bit})$; where $R_{bit}$ = information data rate | |
| Communication System $E_b/N_o$ for the Downlink: | 13.4 | dB | $E_b/N_o = S/N_{oG} - 10logR_{bit}$ | |
| Demodulation Method Seleted: | BPSK | | Values selected in "Modulation-Demodulation W/S, Cell [E30] | |
| Forward Error Correction Coding Used: | R.S. (255, 223) | | Value selected in "Modulation-Demodulation" W/S, also Cell [E30] | |
| System Allowed or Specified Bit-Error-Rate (BER): | 1.0E-05 | | The selected value is transferred from the "Modulation-Demodulation W/S, Cells [E33:E51] | |
| Demodulator Implementation Loss ($L_{DEMOD}$): | 1 | dB | This value is transferred from the "Modulation-Demodulation W/S, Cell [E53] | |
| Communication System Required $E_b/N_o$: | 6.1 | dB | The selected value is transferred from the "Modulation-Demodulation W/S, Cells [F33:F51] | |
| $E_b/N_o$ Threshold: | 7.1 | dB | This is the result of the "Modulation-Demodulation" W/S and is transferred from Cell [H32] | |
| **System Link Margin (LM):** | 6.3 | dB | | |
| ***Ground Station Alternative Signal Analysis Method (SNR Computation):*** ***---------- SNR Method ----------*** | | | | |
| Ground Station Antenna Pointing Loss ($L_{G,PT}$): | 0.1 | dB | This value is transferred from "Antenna Pointing Losses" W/S, Cell [K102] | |
| Ground Station Antenna Gain ($G_{G,A}$): | 15.5 | dBi | This value is selected at "Antenna Gain" W/S, Cell [E58] (assuming a yagi antenna) | |
| Ground Station Total Transmission Line Losses ($L_{G,TLtotal}$): | 0.2 | dB | This value is transferred from the "Receivers" W/S, Cell [J124] | |
| Ground Station Effective Noise Temperature ($T_{G,Neff}$): | 1597 | K | This value is calculated in the "Receivers" W/S and Transferred from Cell [J147] | |
| Ground Station Figure of Merrit ($G/T_G$): | -16.7 | dB/K | $G/T_G = G_{G,A} - L_{G,TLtotal} - 10log(T_{G,Neff})$. This is the ultimate measure of the receiver's performance. | |
| Signal Power at Ground Station LNA Input ($S_{G,LNAin}$): | -152.3 | dBW | $S_{G,LNAin} = EIRP_{G,R} + G_{G,A} - L_{G,PT} - L_{G,TLtotal}$: This is the signal power that has arrived at the ground station receiver. | |
| Ground Station Receiver Bandwidth ($B_{G,RCVR}$): | 2,400 | Hz | Signal Spectrum Must Pass Through This Data Filter | |
| Ground Station Receiver Noise Power ($N_{G,RCVR} = kTB$): | -162.8 | dBW | $N_{G,RCVR} = k + 10log(T_{G,Neff}) + 10log(B_{G,RCVR})$. This is the total noise power arriving at the ground station receiver. | |
| Signal-to-Noise Power Ratio at Ground Station Rcvr ($S/N_{G,RCVR}$): | 10.4 | dB | $S/N_{G,RCVR} = S_{G,LNAin}$(in dBW) - $N_{G,RCVR}$(in dBW) | |
| Analog or Digital System Required $S/N_{RQD}$: | 2.5 | dB | If system is digital, use values from "Modulation-Demodulation" W/S. If analog, use appropriate value from text book. | |
| **System Link Margin (LM):** | 7.9 | dB | LM = $S/N_{G,RCVR} - S/N_{RQD}$ | |

**Table 4: Fox-1E TLM/EXP Downlink Channel Link Budget**

antennas used by ground stations. Given these operational conditions, the TLM/EXP Downlink Channel Link Budget Margin prediction calculation is shown in Table 4. It should be noted that compared with all previous Fox-1x Cubesat VHF downlinks, the Fox-1E UHF downlink results in an additional path loss of approximately 9.5 dB. Regardless, using the $E_b/N_o$ computational method, the resulting link margin is **6.3 dB**. Employing the alternate SNR computational method yields **7.9 dB** of link margin over the maximum slant range.

**Telecommand Uplink Signal Calculation** – The Fox-1E linear transponder design also features a 1200 bps telecommand uplink as part of the Improved Command Receiver (ICR). The exact operational, performance, and design details of the ICR are considered **'sensitive information'** due to its critical capabilities to control the spacecraft. Therefore, without showing the Telecommand Uplink Channel Link Budget spreadsheet, is should be noted that there was one significant operational control parameter change. That assumption was that an AMSAT authorized control ground station would be using 15W of output power to ensure positive control of the spacecraft. Therefore, given these conditions, the 'bottom-line' results can be stated. Again, using the $E_b/N_o$ computational method, the resulting link margin is **28.6 dB**, and employing the alternate SNR computational method yields **13.4 dB** of link margin.

**Performance Parameters** – Having shown in Table 3 that the Fox-1E's combined uplink/downlink SNR under worst case (1° elevation) is approximately **9.1 dB**, it is feasible to then complete a Desirement List of 'hoped for' performance parameters for the three main Linear Transponder functional areas: (1) the TX signal chain; (2) the RX signal chain; and (3) the Improved Command Receiver (ICR)/Interface. Tables 5 to 7 show these initial signal chain performance 'Desirement' lists.

| PERFORMANCE PARAMETERS | REQUIRED VALUES (actually, "DESIREMENTS" pending relaxation) | ESTIMATES (via Analysis or Prototype Measurement) | ACTUALS (as measured) | COMMENTS |
|---|---|---|---|---|
| UHF Downlink Transmit Frequency Range | 435.770 – 435.800 MHz | | | Linear Signal Repeater |
| Maximum Output Power | 500 mW (27 dBm) | | | Transponder (SSB) |
| TLM Channel Frequency | 435.750 MHz | | | -20°C to +50°C |
| Telemetry Channel Maximum Output Power | 100 mW (20 dBm) | | | (nominal pwr shared with transponder users) |
| Frequency Stability (L.O.) | ±5 ppm | | | |
| Spurious Products | >-20 dBc | | | |
| Component Temp. Range | -40°C to +85°C | | | |
| DC Power (Maximum) | 1.0 W (200 mA @ 5.0V) | | | Transmitter power is dynamic and depends on TX loading |
| Third Order Output Intercept Point (OIP3) | >+30 dBm | | | |

**Table 5: Fox-1E Linear Transponder Transmit Signal Chain Performance Values**

| PERFORMANCE PARAMETERS | REQUIRED VALUES (actually, "DESIREMENTS" pending relaxation) | ESTIMATES (via Analysis or Prototype Measurement) | ACTUALS (as measured) | COMMENTS |
|---|---|---|---|---|
| VHF Uplink (Transponder) Receive Frequency Range | 145.870 to 145.900 MHz | | | SSB and CW signals |
| Transponder Bandwidth | 30 kHz (3 dB) | | | |
| Minimum Discernable Signal (MDS)* | -120 dBm | | | |
| Frequency Stability (L.O.) | ±5 ppm | | | -20°C to +50°C |
| Noise Figure (NF) | <3.0 dB | | | |
| Third Order Output Intercept Point (OIP3) | >5 dBm | | | |
| Image Rejection | >45 dB | | | |
| IMD Product Levels | >-50 dBc | | | |
| Component Temp. Range | -40°C to +85°C | | | |
| DC Power (Maximum) | 300 mW (85 mA@3.6V) | | | |

* **for** 10dB S/N SSB signals

**Table 6: Fox-1E Linear Transponder Receive Signal Chain Performance Values**

| PERFORMANCE PARAMETERS | REQUIRED VALUES (actually, "DESIREMENTS" pending relaxation) | ESTIMATES (via Analysis or Prototype Measurement) | ACTUALS (as measured) | COMMENTS |
|---|---|---|---|---|
| Command and Control Receive Channel | XXXX | | | "Protected" Information – see LT RX.ICR Design Doc. |
| Minimum Discernable Signal (MDS)* | XXXX | | | " |
| Channel Bandwidth | XXXX | | | " |
| Noise Figure (NF) | XXXX | | | " |
| Component Temp. Range | -40°C to +85°C | | | |
| DC Power (Maximum) | 200 mW (55 mA @ 3.6 V) | | | |

* for 12dB SINAD

**Table 7: Fox-1E Linear Transponder Improved Command Receiver (ICR) Performance Values**

**COMPONENT SELECTION** – An RF architecture, a transmit/receive frequency plan, link calculations, and initial signal chain performance parameter (desirements) lists provide the foundation for initial component selection. AMSAT satellites are built with available, off-the-shelf components. Component selection is accomplished considering the following eight (8) principal considerations (not in priority order):

1. SWAP – CubeSat regulated size and weight constraints, as well as power limitations of on-orbit power generation and energy storage, i.e., solar panels and battery capacity, significantly affect design alternatives. Consequently, component Size, Weight And Power (SWAP) are premier considerations when designing satellite subsystems.
2. Performance – High-performance components are of paramount importance, EXCEPT where such performance comes at a price of high complexity and/or can only be realized at the cost of high power, limited temperature range, reliability, size, or weight. Such trades usually make the acceptance of high performance components unacceptable for AMSAT satellite use.
3. Availability/Delivery – Facing usually tight time lines for assembly, test, qualification and launch, Just-in-time (JIT) assembly, repair, and troubleshooting demand readily available components from distributors who can ship on short notification.
4. Temperature – Accommodating extreme operating temperatures is a critical part of space electronics design. Consequently, parts selection first looks at temperature ranges of -40°C to +85°C for all components, especially those in critical "must-survive" circuitry.
5. Packaging – SWAP considerations dictate that most satellite components be packaged via surface mount technologies. However, ultra-small parts requiring advanced technology assembly capabilities and any type of Grid Array (xGA) are generally NOT acceptable under the cost-effective manufacturing available to AMSAT. Rather, typical component packages used include small outline transistors (SOT), small outline packages (xSOP), flat packs (leaded and non-leaded), etc.
6. Reliability – Space mission success depends heavily on high-reliability parts. Therefore, components manufactured by reputable IC manufacturers with proven active and passive device processes are preferred. Most components are of commercial high grade with limited reliability data available. However, where applicable reliability data is available, it is analyzed for space operations applicability.
7. Radiation Tolerance – Although AMSAT does not purchase Rad-Hardened parts, through radiation part data bases, parts selection favors critical parts with radiation-tolerance.

However, more often, radiation concerns are mitigated by the application of tantalum protection/shielding.

8. Cost – AMSAT satellite component purchases are generally in small lots (≤ 10 pieces). Consequently, large volume price breaks are generally not realizable. Often, smaller quantity piece part buys occur. Therefore, cost can't be considered a paramount consideration. Additionally, given considerable tight size, weight, and power (SWAP) constraints in CubeSat design, extra cost for special components is often an acceptable trade.

Figure 7 is a lower level block diagram of the design stages of the Fox-1E's linear transponder. The figure shows three signal chains: (1) the receive (RX) signal chain, (2) the transmit (TX) signal chain, ad (3) the command and control (C2) - signal chain (with interface circuitry). The initial set of selected components is shown labeled by manufacturer and part name/number.

The design goal is to build the entire linear transponder's circuitry on two (2) PCBs. The figure shows the preliminary design approach is to attempt to fit the receive chain front end components, along with the command and control chain components and interface circuitry, on one PCB. Leaving a second PCB to house the transmit chain components with adequate room for shielding.

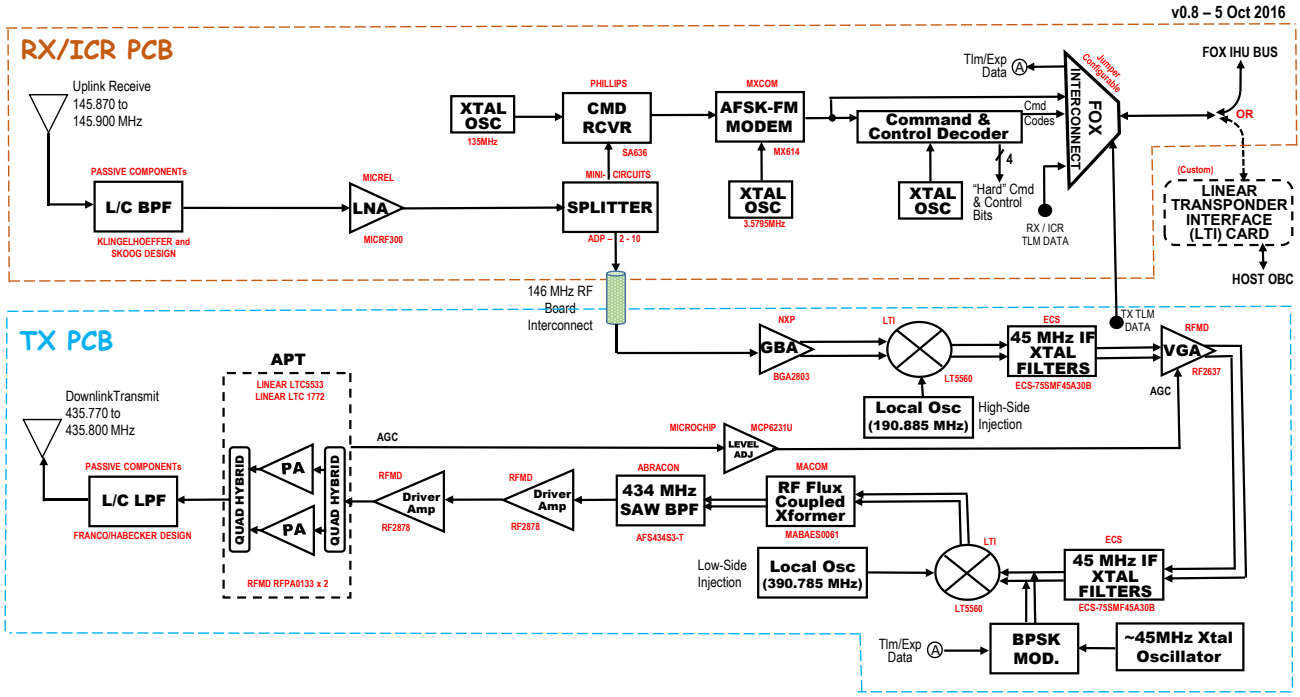



**FIGURE 7: Linear Transponder Signal Chain Cascaded Components Block Diagram**

**RF SIGNAL CHAIN SYSTEM ANALYSIS** – The gap between RF Computer Aided Engineering (CAE) tools at the circuit level and those needed for initial design at the system level is often filled by RF signal chain system analysis spreadsheet tools using common office software such as Microsoft Excel. Signal chain system analysis is regularly employed as an early step in preliminary RF design. Such an analysis facilitates performance evaluation and component selection for the architectures under consideration.

The analysis methodology cascades the specifications of the components in the signal chain to approximate system performance. The linear and nonlinear characteristics of an RF system comprising a cascade of two-port components can be determined such that system characteristics are derived at the system input, system output, and the nodes between the components. It is intended to determine if the proposed system's key performance measures, e.g., noise figure, third-order intercept point, power, Signal-to-Noise Ratio, and others can satisfy the requirements. The analysis is based on using frequency domain characteristics of each top level two-port component

in the RF system design. The collection of the RF performance parameters for each constituent component in the cascade is used to calculate the system performance at each node of the system design.

Myriad spreadsheet template applications have been created to simplify the process of performing an RF signal chain cascade analysis. Unfortunately, regardless of the completeness of the spreadsheet application, an analysis generated by using such spreadsheets does NOT reveal several potential operational circuit abnormalities, e.g.,

- Spurious RF products interfering with circuit operation,
- Impedance mis-match issues,
- Noise problems,
- RF sneak paths,
- Electromagnetic Interference (radiated/conducted), or
- Variation analysis, reliability, environmental stress, etc.

The previously generated Frequency Planning and Spur Web plots are one attempt to address some of these deficiencies. However, in the absence of expensive, complex, steep learning curve electronic design aid (EDA) software, many items on this list will not be settled until prototype builds and stack integration and testing of the initial engineering design models (EDM) of the CubeSat's PCBs have been performed.

As the name "Linear Transponder" denotes, linearity is a major issue in the design. Design performance is typically limited by intermod performance at the top end of the dynamic range and noise at the bottom end. Intermod and noise performance are at odds with one another and many times improving the performance of one results in degradation of the other. The cascaded intercept point is referred to the input (IIP) or output (OIP) for convenience. A higher intercept point yields larger dynamic range. Consequently, intercept points are commonly used as a performance characteristic of RF systems. In general the higher the intercept point, the more tolerant a system is to interference. Transmitters and amplifiers generally have their intercept points referred to the output and receivers have them referred to their input.

Signal Chain analyses yields significant linear transponder design performance information. Tables 8 and 9 display the results for the Fox-1E linear transponder's preliminary receive and transmit signal chain designs.

**Receive Signal Chain** – The calculated receive signal chain performance is shown in Table 8. Each component comprising a stage has data sheet provided performance parameter values inserted for Gain(G)/Insertion Loss(IL), Noise Figure (NF), Output Third Order Intercept Point (OIP3), Output 1dB Compression ($IP_{1dB}$), etc. The Noise Figure for passive elements is equal to their insertion loss and their OIP3 values are set to 100dBm for the purpose of more accurate calculations.

By examining two classic cascade performance equations, the FIRST COMPONENT in the signal chain can be shown to dominate the Noise Factor/Figure, while the LAST COMPONENT dominates the Intercept Point. Therefore, for an optimum signal chain design, the initial (low noise) amplifier (LNA) should exhibit minimum noise and the last amplifier should exhibit a high Intercept Point value.

*Cascaded Noise Figure Equation* - The (total) Noise Factor ($F_{total}$) for an N-stage cascade receiver can be calculated from the well-known Friis equation:

$$\mathbf{F_{total} = F_1 + (F_2 - 1)/G_1 + (F_3 - 1)/G_1G_2 + (F_4 - 1)/G_1G_2G_3 + .... + (F_N - 1)/G_1G_2....G_{N-1}}$$

and the equivalent Noise Figure ($NF_{dB}$) as: $NF_{dB} = 10 \log F$

From the Friis equation, it can be seen that the initial component in the cascade effectively establishes the Noise Factor/Figure for the entire receive signal chain. This is because the (noise) contribution of the latter components in the chain is diminished due to their division by

the preceding stages' cumulative gain factors ($G_i$). Consequently, it is desirable to select an initial (active) component that exhibits a small Noise Factor/Figure. This first (active) component is generally a Low Noise Amplifier (LNA).

Further along the signal chain, an amplifier and mixer are generally cascaded. Examining the Friis equation, it can be seen that OPTIMALLY the amplifier should precede the mixer to MINIMIZE the overall Noise Factor/Figure.

*Cascaded 3rd Order IMD Intercept Point Equation* - If the intermod-generated products throughout the cascaded signal chain are assumed to be in phase, then coherent addition should be used. This will yield a worst case intercept point. Examining the third order output intercept point (OIP3) equation for an N-stage receive signal chain,

$$\mathbf{OIP3_{total} = \{1/(OIP3_1G_{2....}G_{N-1}G_N) + \ldots + 1/(OIP3_{N-2}G_{N-1}G_N) + 1/(OIP3_{N-1}G_N) + 1/(OIP3_N)\}^{-1}}$$

It can be seen that the opposite component arrangement, i.e., mixer preceding an amplifier, would MAXIMIZE the overall third order intercept point. Hence, MINIMIZING gain is desirable so that the OIP3 may be MAXIMIZED. This is the basic trade-off between receive signal chain design for Noise and Intercept Points.

These two equations motivate one of the most important decisions that must be made in the design of a receive signal chain-determining gain distribution. This distribution determines the Noise Figure and the signal levels at various stages in the chain. To obtain the best NF, adequate gain is required prior to the first mixer stage. Mixers tend to have poor NFs and a mixer design with the lowest spurious responses may well exhibit a high loss. However, minimum intermod (IM) product levels occur when the signal level is as low as possible prior to final channel bandwidth selection. Usually this implies minimum gain prior to the final IF amplifier, where channel bandwidths are likely to be established. In some cases it may be necessary to accept lower sensitivity to avoid high spurious response and IM levels.

Therefore, receive signal chain design is an interactive process centered around signal chain analysis. Initial component selections are made; the NF, IPs, and levels of spurious response closest to the RF/IF are evaluated and compared to the "desirements." This then leads to a second round of component selections, more analysis, and so on until the appropriate compromise (trade-offs) have been achieved.

| FOX-1E | Title: Fox-1E U/V LINEAR TRANSPONDER (Receive Chain) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Filename: Linear Transponder Cascade Analysis (v0.3) | | | | | | | | |
| | | | | | | | | | | COMMENTS |
| Stage Name | | LC BPF | LNA | SAW BPF | SPLITTER | AMP | MIXER | XTAL FILTER | $VGA_{AGC}$ | |
| Manufacturer | | J.K. | MICREL | Golledge | Mini-Circuits | NXP | CEL | ECS | RFMD | |
| Part ID | | LC (ELSIE) | MICRF 300YC6 | TA146 | MCL ADP-2-10 | BGA2803 | uP2757TB | 75SMF45A30B | RF2637 | |
| Transducer Power Gain, $G_i$ (dB) | | -0.96 | 18.5 | -6.5 | -3.3 | 22.5 | 16.0 | -3.0 | 10.0 | $V_{GC}$ = 1.5 Volts |
| G Hot (dB) | | | 18.5 | | | 21.5 | | | 8.0 | |
| G Cold (dB) | | | 18.4 | | | 25.1 | | | 11.0 | |
| Noise Figure, $NF_i$ (dB) | | 0.96 | 1.2 | 6.5 | 3.3 | 3.6 | 9.0 | 3.0 | 4.4 | |
| NF, Hot (dB) | | | 1.3 | | | 4.5 | | | 5.4 | |
| NF, Cold (dB) | | | 1.2 | | | 3.0 | | | 3.0 | |
| Output $IP3_i$ (dBm) | | 100.0 | 3.5 | 100.0 | 100.0 | 6.0 | 5.0 | 27.0 | -8.0 | |
| OIP3, Hot (dBm) | | | 5.5 | | | 5.0 | | | -9.0 | |
| OIP3, Cold (dBm) | | | 0.5 | | | 7.0 | | | -10.0 | |
| Output $P1dB_i$ (dBm) | | 90.0 | -8.5 | 90.0 | 90.0 | -6.0 | -3.0 | 17.0 | -12.0 | |
| OP1dB, Hot (dBm) | | | -6.5 | | | -6.0 | | | -13.0 | |
| OP1dB, Cold (dBm) | | | -10.0 | | | -5.0 | | | -10.0 | |
| Maximum Input Power, $P_{INmax}$ (dBm) | | 27.0 | 10.0 | 10.0 | 27.0 | 10.0 | -10.0 | 27.0 | 10.0 | |
| Nominal Supply Voltage (V) | | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | |
| Nominal Device Bias Current (mA) | | 0.0 | 2.5 | 0.0 | 0.1 | 5.7 | 5.6 | 1.0 | 11.5 | |
| +25 °C Calculated Values | | | | | | | | | | EQUATIONS |
| Signal Power Out $P_{OUT}$ (dBm) @ stage i | | -121.0 | -102.5 | -109.0 | -112.3 | -89.8 | -73.8 | -76.8 | -66.8 | $P_{OUT_i} = P_{OUT_{i-1}} + G_i$ |
| Input Signal Level $S_{IN}$ (dBm) = | -120.0 | | | | | | | | | |
| Cumulative Gain $G_{CUM}$ (dB) | 0.0 | -1.0 | 17.5 | 11.0 | 7.7 | 30.2 | 46.2 | 43.2 | 53.2 | $G_{CUM_i} = G_{CUM_{i-1}} + G_i$ |
| | | | | | | | | | | |
| Stage Noise Temperature $T_{STAGE\,i}$ (°K) | | 71.7 | 109.7 | 17.7 | 26.0 | 63.0 | 1.9 | 0.0 | 0.0 | $T_{STAGE_i} = T_o\{(10^{NF_i/10} - 1)/(10^{G_{CUM_{i-1}}/10})\}$ & $T_o$ = 290 °K |
| Cumulative Noise Temperature $T_{CUM\,i}$ (°K) | | 71.7 | 181.4 | 199.1 | 225.1 | 288.1 | 290.0 | 290.0 | 290.0 | $T_{CUM_i} = T_{STAGE_i} + T_{CUM_{i-1}}$ |
| Cumulative Noise Figure $NF_{CUM\,i}$ (dB) | | 1.0 | 2.1 | 2.3 | 2.5 | 3.0 | 3.0 | 3.0 | **3.0** | $NF_{CUM_i} = 10\log(\frac{T_{CUM_i}}{T_o} + 1)$ |
| | | | | | | | | | | |
| Stage Input Intercept Point $IIP3_i$ (dBm) | | 101.0 | -15.0 | 106.5 | 103.3 | -16.5 | -11.0 | 30.0 | -18.0 | $IIP3_{STAGE_i} = OIP3_i - G_i$ |
| Cumulative Input Intercept Point $IIP3_{CUM\,i}$ (dBm) | 100.0 | 97.4 | -14.0 | -14.0 | -14.0 | -24.6 | -41.3 | -41.3 | -61.3 | $IIP3_{CUM_i} = -10\log[10^{\frac{-IIP3_{CUM_{i-1}}}{10}} + 10^{\frac{G_{CUM_{i-1}} - IIP3_{STAGE_i}}{10}}]$ |
| Cumulative Output Intercept Point $OIP3_{CUM\,i}$ (dBm) | | 96.5 | 3.5 | -3.0 | -6.3 | 5.6 | 4.9 | 1.9 | -8.0 | $OIP3_{CUM_i} = IIP3_{CUM_i} + G_{CUM_i}$ |
| Output Inter-Mod Product IM3 @ Stage i (dBc) | | -434.9 | -211.9 | -211.9 | -211.9 | -190.7 | -157.3 | -157.3 | -117.4 | $IM3_{STAGE_i} = 2(P_{OUT_i} - OIP3_{CUM_i})$ |
| | | | | | | | | | | |
| Stage Compression Input Point $IP_{1dB}$ (dBm) | | 92.0 | -26.0 | 97.5 | 94.3 | -27.5 | -18.0 | 21.0 | -21.0 | $IP_{1dB_{STAGE\,i}} = OP_{1dB_i} - G_i + 1_{dB}$ |
| Cumulative Compression Input Point $IP_{1dBCUM}$ (dBm) | 100.0 | 91.3 | -25.0 | -25.0 | -25.0 | -35.6 | -48.5 | -48.5 | -64.4 | $IP_{1dB_{CUM_i}} = -10\log[10^{\frac{-IP_{1dB_{CUM_{i-1}}}}{10}} + 10^{\frac{G_{CUM_{i-1}} - IP_{1dB_{STAGE_i}}}{10}}]$ |
| Cumulative Compression Output Point $OP_{1dBCUM}$ (dBm) | | 89.4 | -8.5 | -15.0 | -18.3 | -6.4 | -3.2 | -6.3 | -12.1 | $OP_{1dB_{CUM\,i}} = IP_{1dB_{CUM\,i}} + G_{CUM\,i} - 1_{dB}$ |
| DC Power Input | | | | | | | | | | Total DC Input Power (mW) |
| Power, mW | | 0 | 7.5 | 0 | 0.3 | 17.1 | 16.8 | 3 | 34.5 | $P_{TOTAL} = \sum_{i=1}^{N} P_i$ **79.2** |

**Table 8: Fox-1E Linear Transponder Receive Signal Chain Analysis**

Table 8 shows that the predicted cumulative gain (dB), noise figure (dB), and third order intermodulation product levels (dBc) performance values for the cascade receive chain of Figure 7 are acceptable. However, this same cascade also exhibits disappointing cumulative Third Order Input Intercept Point ($IIP3_{CUM}$) and cumulative Input/Output Compression Point (IP1dB/OP1dB) values. This aspect of the initial design requires further analysis, and highlights the tradeoffs that have been made.

**Transmit Signal Chain** – The transmit signal chain performance of the Fox-1E initial Linear Transponder is shown in Table 9 below.

| FOX-1E | Title: Fox-1E U/V LINEAR TRANSPONDER (Transmit Chain) | | | | | | Reference: Component Data Sheets | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Filename: Linear Transponder Cascade Analysis (v0.3) | | | | | | Last Updated: 4 July 2016 | | | |
| | | | | | | | | | | COMMENTS |
| Stage Name | COMBINER | MIXER | SAW BPF | AMP | SAW BPF | DRIVER AMP | DRIVER AMP | PWR AMP | LC LPF | |
| Manufacturer | Mini-Circuits | CEL | Golledge | NXP | Golledge | RFMD | RFMD | RFMD | M.F. & D.H. | |
| Part ID | MCL ADP-2-10 | uPC8106TB | TA1408A | BGA2803 | TA1408A | RF2878 | RF2878 | RPA0133 | LC (ELSIE) | |
| Transducer Power Gain, $G_i$ (dB) | -3.3 | 9.5 | -2.0 | 22.5 | -2.0 | 21.0 | 21.0 | 25.0 | -1.0 | |
| G Hot (dB) | | | | 21.5 | | | | | | |
| G Cold (dB) | | | | 25.1 | | | | | | |
| Noise Figure, $NF_i$ (dB) | 3.3 | 8.5 | 2.0 | 3.6 | 2.0 | 1.9 | 1.9 | 5.4 | 1.0 | |
| NF, Hot (dB) | | | | 4.5 | | | | | | |
| NF, Cold (dB) | | | | 3.0 | | | | | | |
| Output $IP3_i$ (dBm) | 100.0 | 5.5 | 100.0 | 6.0 | 100.0 | 29.0 | 29.0 | 35.0 | 100.0 | |
| OIP3, Hot (dBm) | | | | 5.0 | | | | | | |
| OIP3, Cold (dBm) | | | | 7.0 | | | | | | |
| Output $P1dB_i$ (dBm) | 90.0 | -8.0 | 27.0 | -6.0 | 27.0 | 12.5 | 12.5 | 27.0 | 90.0 | |
| OP1dB, Hot (dBm) | | | | -6.0 | | | | | | |
| OP1dB, Cold (dBm) | | | | -5.0 | | | | | | |
| Maximum Input Power, $P_{INmax}$ (dBm) | 27.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 30.0 | |
| Nominal Supply Voltage (V) | 0.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.0 | 3.6 | 0.0 | |
| Nominal Device Bias Current (mA) | 0.0 | 9.0 | 0.0 | 5.7 | 0.0 | 22.0 | 22.0 | 150.0 | 0.0 | |
| **+25 °C Calculated Values** | | | | | | | | | | **EQUATIONS** |
| Signal Power Out $P_{OUT}$ (dBm) @ stage i | -70.1 | -60.6 | -62.6 | -40.1 | -42.1 | -21.1 | -0.1 | 24.9 | 23.9 | $P_{OUT_i} = P_{OUT_{i-1}} + G_i$ |
| Receive IF Input Signal Level $S_{IN}$ (dBm) = -66.8 | | | | | | | | | | |
| Cumulative Gain $G_{CUM}$ (dB) | -3.3 | 6.2 | 4.2 | 26.7 | 24.7 | 45.7 | 66.7 | 91.7 | 90.7 | $G_{CUM_i} = G_{CUM_{i-1}} + G_i$ |
| | | | | | | | | | | |
| Stage Noise Temperature $T_{STAGE\,i}$ (°K) 0.0 | 330.0 | 3769.3 | 40.7 | 142.3 | 0.4 | 0.5 | 0.0 | 0.0 | 0.0 | $T_{STAGE_i} = T_o\{(10^{NF_i/10} - 1)/(10^{G_{CUM_{i-1}}/10})\}$ & $T_o$ = 290 °K |
| Cumulative Noise Temperature $T_{CUM\,i}$ (°K) | 330.0 | 4099.3 | 4140.0 | 4282.3 | 4282.7 | 4283.2 | 4283.2 | 4283.2 | 4283.2 | $T_{CUM_i} = T_{STAGE_i} + T_{CUM_{i-1}}$ |
| Cumulative Noise Figure $NF_{CUM\,i}$ (dB) | 3.3 | 11.8 | 11.8 | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 | 12.0 | $NF_{CUM_i} = 10\log(\frac{T_{CUM_i}}{T_o} + 1)$ |
| | | | | | | | | | | |
| Stage Input Intercept Point $IIP3_i$ (dBm) | 103.3 | -4.0 | 102.0 | -16.5 | 102.0 | 8.0 | 8.0 | 10.0 | 101.0 | $IIP3_{STAGE_i} = OIP3_i - G_i$ |
| Cumulative Input Intercept Point $IIP3_{CUM\,i}$ (dBm) 100.0 | 98.3 | -0.7 | -0.7 | -20.7 | -20.7 | -22.2 | -37.8 | -56.8 | -56.8 | $IIP3_{CUM_i} = -10\log[10^{\frac{-IIP3_{CUM_{i-1}}}{10}} + 10^{\frac{G_{CUM_{i-1}} - IIP3_{STAGE_i}}{10}}]$ |
| Cumulative Output Intercept Point $OIP3_{CUM\,i}$ (dBm) 100.0 | 95.0 | 5.5 | 3.5 | 6.0 | 4.0 | 23.5 | 28.9 | 34.9 | 33.9 | $OIP3_{CUM_i} = IIP3_{CUM_i} + G_{CUM_i}$ |
| Output Inter-Mod Product IM3 @ Stage i (dBc) | -330.3 | -132.2 | -132.2 | -92.1 | -92.1 | -89.2 | -58.0 | -20.1 | -20.1 | $IM3_{STAGE_i} = 2(P_{OUT_i} - OIP3_{CUM_i})$ |
| | | | | | | | | | | |
| Stage Compression Input Point $IP_{1dB}$ (dBm) | 94.3 | -16.5 | 30.0 | -27.5 | 30.0 | -7.5 | -7.5 | 3.0 | 92.0 | $IP_{1dB_{STAGE\,i}} = OP_{1dB_i} - G_i + 1_{dB}$ |
| Cumulative Compression Input Point $IP_{1dBCUM}$ (dBm) 100.0 | 93.3 | -13.2 | -13.2 | -31.8 | -31.8 | -35.0 | -53.3 | -64.1 | -64.1 | $IP_{1dB_{CUM_i}} = -10\log[10^{\frac{-IP_{1dB_{CUM_{i-1}}}}{10}} + 10^{\frac{G_{CUM_{i-1}} - IP_{1dB_{STAGE_i}}}{10}}]$ |
| Cumulative Compression Output Point $OP_{1dBCUM}$ (dBm) 100.0 | 89.0 | -8.0 | -10.0 | -6.1 | -8.1 | 9.7 | 12.4 | 26.6 | 25.6 | $OP_{1dB_{CUM\,i}} = IP_{1dB_{CUM\,i}} + G_{CUM\,i} - 1_{dB}$ |
| **DC Power Input** | | | | | | | | | | **Total DC Input Power (mW)** |
| Power, mW | 0 | 27 | 0 | 17.1 | 0 | 66 | 66 | 540 | | $P_{TOTAL} = \sum_{i=1}^{N} P_i$ **176.1** |

**Table 9: Fox-1E Linear Transponder Transmit Signal Chain Analysis**

Examination of Table 9 verifies that by adding a second Power Amplifier driver stage, the desired output power gain is achievable. This end-stage gain is realized through a total TX cumulative gain of approximately 91 dB! The transmit signal chain's Noise Figure is a manageable 12 dB and the Intermodulation products generated are >20 dBc. Similarly to the RX signal chain performance, the Third-Order Output Intercept Point (OIP3) and the Compression Output Point ($OP_{1dB}$) are decent values, but would seem to be at the (tradeoff) expense of the Third-order Input Intercept Point (IIP3) and the Compression Input Point ($IP_{1dB}$). Furthermore, the $OP_{1dB}$ value of 25.6 dBm is barely above the output of the final stage Power Amplifier at 23.9 dBm – this would seem to be much too close to compression, although this does not account for the effects of the Average Power Tracking (APT) circuitry.

**NEXT STEPS –** This Fox-1E Linear Transponder Communications Performance Analysis document contains most of the basic assumptions and decisions that underpin the Linear Transponder design at this stage of the development. It is expected that the AMSAT RF Engineering team will critically review this entire document. Their comments, changes/revisions and suggestions for update will then be incorporated. Particularly, the current set of components being used in the design of the RX, TX, and ICR prototype units needs to be updated and reflected in Figure 7 and the signal chain performance values for those functional areas, i.e., Tables 8 and 9, need to then be rerun. Lastly, the "Desirements" List Performance Values in Tables 5, 6, and 7 must be reviewed for feasibility and updated to reflect initial prototype experience.

**Bottom Line: ALL of the assumptions, calculations and results require continued validation and more investigation/study.**

The review and update of this document can proceed in parallel with continued prototype development and on into the PDR/CDR progress evaluation period. However, at the point of final prototype prove-out, functional unit integration, and performance measurements, this document should reflect the performance of the proven design and form the basis for continuing into Engineering Model (EM) assembly and test. Thus this Linear Transponder Comm. Performance Analysis should be considered a 'living document' subject to change as the Fox-1E RF Engineering team evolves the design into the final Flight Model (FM) unit.

## FOX-1E DOWNLINK SPECIFICATION

// This is the master copy of the downlink structures. A program called FoxStruct.py will turn this into a structure for the flight software.
// A different Python script can also be used to make tables for FoxTelem. My intent is that this spreadsheet can also be included
// into a formal document defining the downlink more completely.
// The format must be as follows: 1) A comment (// in column A) can appear anywhere. 2) A structure must start with a column A that has
// "Structure:name,file:filename". The structure must end with "END". For flight software, the yellow highlighted columns are used.
Version: 2.4

Structure: header_t, file:headerDownlink.h

| // | Field Name | Code Name | Size (Bits) | Type | C Type | Conversion | Label | Description | Bit Offset | Byte Offset | Short Offset | Long offset | Module |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FoxID | oldFoxID | 3 | Unsigned | | 0 | | 0x01 specifies Fox-1A (each Fox satellite will have a unique ID) | 0 | 0 | 0 | 0 | |
| | resetCnt | resetCnt | 16 | Unsigned | | 1 | Resets | Total number of times IHU has reset since initial on-orbit startup | 3 | 0.375 | 0.1875 | 0.09375 | |
| | uptime | uptime | 25 | Unsigned | | 1 | Sec | This is the IHU uptime in seconds since the last reset | 19 | 2.375 | 1.1875 | 0.59375 | |
| | type | type | 4 | Unsigned | | 0 | | This identifies the payload type | 44 | 5.5 | 2.75 | 1.375 | |
| | Extended ID | FoxID | 8 | | | | | | 48 | 6 | 3 | 1.5 | |
| | Downlink Spec Version | majorVersion | 3 | | | | | Major version—changes are incompatible | 56 | 7 | 3.5 | 1.75 | |
| | Downlink Spec Version | minorVersion | 5 | | | | | Minor version—changes are compatible | 59 | 7.375 | 3.6875 | 1.84375 | |
| END | | | | | | | | Total length | 64 | 8 | 4 | 2 | |

//
Structure: commonPayload_t,file:commonDownlink.h

| // | Field Name | Code Name | Size (Bits) | Type | C Type | Conversion | Label | Description | Bit Offset | Byte Offset | Short Offset | Long offset | Module |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cell A | battAv | 12 | Unsigned | | 4 | V | Battery pair A voltage (0-2.5V scale) | 0 | 0 | 0 | 0 | Battery |
| | Cell A + B | battBv | 12 | Unsigned | | 4 | V | Battery pairs A+B voltage (0-3.3V scale) | 12 | 1.5 | 0.75 | 0.375 | Battery |
| | Cell A + B + C | battCv | 12 | Unsigned | | 4 | V | Battery pairs A+B+C voltage (0-5.0V scale) | 24 | 3 | 1.5 | 0.75 | Battery |
| | Temperature A | battAtemp | 12 | Unsigned | | 8 | C | Battery pair A temperature | 36 | 4.5 | 2.25 | 1.125 | Battery |
| | Temperature B | battBtemp | 12 | Unsigned | | 8 | C | Battery pair B temperature | 48 | 6 | 3 | 1.5 | Battery |
| | Temperature C | battCtemp | 12 | Unsigned | | 8 | C | Battery pair C temperature | 60 | 7.5 | 3.75 | 1.875 | Battery |
| | Current | battCurr | 12 | Signed | | 9 | mA | Total Battery DC current | 72 | 9 | 4.5 | 2.25 | Battery |
| | Board Temp | battCardTemp | 12 | Unsigned | | 8 | C | Battery Card Temperature | 84 | 10.5 | 5.25 | 2.625 | Battery |
| | Voltage | posXv | 12 | Unsigned | | 23 | V | +X solar Panel voltage | 96 | 12 | 6 | 3 | +X Panel |
| | Voltage | posYv | 12 | Unsigned | | 23 | V | +Y solar Panel voltage | 108 | 13.5 | 6.75 | 3.375 | +Y Panel |
| | Voltage | posZv | 12 | Unsigned | | 23 | V | +Z solar Panel voltage | 120 | 15 | 7.5 | 3.75 | +Z Panel |
| | Voltage | negXv | 12 | Unsigned | | 23 | V | -X solar Panel voltage | 132 | 16.5 | 8.25 | 4.125 | -X Panel |
| | Voltage | negYv | 12 | Unsigned | | 23 | V | -Y solar Panel voltage | 144 | 18 | 9 | 4.5 | -Y Panel |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Voltage | negZv | 12 | Unsigned | | 23 | V | -Z solar Panel voltage | 156 | 19.5 | 9.75 | 4.875 | -Z Panel |
| | Temp | posXt | 12 | Unsigned | | 24 | C | +X solar Panel temperature | 168 | 21 | 10.5 | 5.25 | +X Panel |
| | Temp | posYt | 12 | Unsigned | | 24 | C | +Y solar Panel temperature | 180 | 22.5 | 11.25 | 5.625 | +Y Panel |
| | Temp | posZt | 12 | Unsigned | | 24 | C | +Z solar Panel temperature | 192 | 24 | 12 | 6 | +Z Panel |
| | Temp | negXt | 12 | Unsigned | | 24 | C | -X solar Panel temperature | 204 | 25.5 | 12.75 | 6.375 | -X Panel |
| | Temp | negYt | 12 | Unsigned | | 24 | C | -Y solar Panel temperature | 216 | 27 | 13.5 | 6.75 | -Y Panel |
| | Temp | negZt | 12 | Unsigned | | 24 | C | -Z solar Panel temperature | 228 | 28.5 | 14.25 | 7.125 | -Z Panel |
| | Board Temp | PSUTemp | 12 | Unsigned | | 8 | C | Power Supply Card Temperature | 240 | 30 | 15 | 7.5 | MPPT |
| | Spacecraft Spin | spin | 12 | Signed | | 12 | rpm | Calculated spin rate using solar cells | 252 | 31.5 | 15.75 | 7.875 | Computer Software |
| | PA Current | TxPAi | 12 | Unsigned | | 10 | mA | Transmit power amplifier current | 264 | 33 | 16.5 | 8.25 | Radio |
| | TX Temperature | TxTemp | 12 | Unsigned | | 7 | C | Transmitter card temperature | 276 | 34.5 | 17.25 | 8.625 | Radio |
| | Fwd Power | FwdPower | 12 | Unsigned | | 0 | mW | Receiver card temperature | 288 | 36 | 18 | 9 | Radio |
| | RSSI | rssi | 12 | Unsigned | | 14 | dBm | Received Signal Strength Indication | 300 | 37.5 | 18.75 | 9.375 | Radio |
| | Temperature | IHUcpuTemp | 12 | Unsigned | | 15 | C | Internal Temperature of IHU | 312 | 39 | 19.5 | 9.75 | Computer Hardware |
| | Rotation | Xvelocity | 12 | Unsigned | | 13 | dps | Angular Veolcity around X Axis | 324 | 40.5 | 20.25 | 10.125 | +X Panel |
| | Rotation | Yvelocity | 12 | Unsigned | | 13 | dps | Angular Veolcity around Y Axis | 336 | 42 | 21 | 10.5 | +Y Panel |
| | Rotation | Zvelocity | 12 | Unsigned | | 13 | dps | Angular Veolcity around Z Axis | 348 | 43.5 | 21.75 | 10.875 | +Z Panel |
| | EXP4 Temp | Exp4Temp | 12 | Unsigned | | 7 | C | Experiment 4 card temperature | 360 | 45 | 22.5 | 11.25 | Experiments |
| | Current | PSUCurrent | 12 | Unsigned | | 22 | mA | Power Supply DC Current | 372 | 46.5 | 23.25 | 11.625 | MPPT |
| | Ref Power | ReflectedPwr | 12 | Unsigned | | 0 | mW | Reflected Power from the Transmit Antenna | 384 | 48 | 24 | 12 | Radio |
| | 3V | ICR3V | 12 | Unsigned | | 23 | V | ICR 3V | 396 | 49.5 | 24.75 | 12.375 | Command Receiver (ICR) |
| | 3V Prot | ICR3VProt | 12 | Unsigned | | 23 | V | ICR 3V Proteted | 408 | 51 | 25.5 | 12.75 | Command Receiver (ICR) |
| | 2.5V | ICR2dot5V | 12 | Unsigned | | 23 | V | ICR 2.5V | 420 | 52.5 | 26.25 | 13.125 | Command Receiver (ICR) |
| | 2.5V Prot | ICR2dot5VProt | 12 | Unsigned | | 23 | V | ICR 2.5V Protected | 432 | 54 | 27 | 13.5 | Command Receiver (ICR) |
| | None | rf6 | 12 | Unsigned | | 0 | - | None | 444 | 55.5 | 27.75 | 13.875 | NONE |
| | None | rf7 | 12 | Unsigned | | 0 | - | None | 456 | 57 | 28.5 | 14.25 | NONE |
| | Sensor Power | MuxTest | 12 | Unsigned | | 23 | V | Sensor Power Voltage at the ICR | 468 | 58.5 | 29.25 | 14.625 | Command Receiver (ICR) |
| | VGA Control | LtVGACtl | 12 | Unsigned | | 23 | V | Control Voltage to the Variable Gain Amplifier (VGA) | 480 | 60 | 30 | 15 | Radio |
| | None | pad | 4 | Unsigned | | 34 | - | Unused | 492 | 61.5 | 30.75 | 15.375 | NONE |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| END | | | | | | | | Totals: | 496 | 62 | 31 | 15.5 | |
| | | | | | | | | | | | | | |
| Structure: realtimeSpecific_t, file:realtimeDownlink.h | | | | | | | | | | | | | |
| | Diagnostic Info | IHUdiagData | 32 | Unsigned | DownlinkIHUInfo | 18 | - | Diagnostic Data on IHU Performance | 496 | 62 | 31 | 15.5 | Computer Software |
| | Vanderbilt Radiation | expFailure0 | 1 | Unsigned | | 17 | - | Vanderbilt University Vulcan Experiment Indicator | 528 | 66 | 33 | 16.5 | Experiments |
| | Exp 1 | expFailure1 | 1 | Unsigned | | 17 | - | Experiment 1 Failure Indicator | 529 | 66.125 | 33.0625 | 16.53125 | Experiments |
| | Exp 2 | expFailure2 | 1 | Unsigned | | 17 | - | Experiment 2 Failure Indicator | 530 | 66.25 | 33.125 | 16.5625 | Experiments |
| | Exp 3 | expFailure3 | 1 | Unsigned | | 17 | - | Experiment 3 Failure Indicator | 531 | 66.375 | 33.1875 | 16.59375 | Experiments |
| | Battery I2C | I2CfailureBatt | 1 | Unsigned | | 17 | - | Battery I2C Bus failure indicator | 532 | 66.5 | 33.25 | 16.625 | Computer Hardware |
| | PSU1 I2C | I2CfailurePSU1 | 1 | Unsigned | | 17 | - | PSU2 I2C Bus failure indicator | 533 | 66.625 | 33.3125 | 16.65625 | Computer Hardware |
| | PSU2 I2C | I2CfailurePSU2 | 1 | Unsigned | | 17 | - | PSU2 I2C Bus failure indicator | 534 | 66.75 | 33.375 | 16.6875 | Computer Hardware |
| | Ground Resets | TLMresets | 4 | Unsigned | | 1 | - | Number of times command stations reset stored telemetry | 535 | 66.875 | 33.4375 | 16.71875 | Computer Hardware |
| | RX Antenna | RxAntDeploy | 1 | Unsigned | | 16 | - | 2m Antenna status | 539 | 67.375 | 33.6875 | 16.84375 | Radio |
| | TX Antenna | TxAntDeploy | 2 | Unsigned | | 16 | - | 70cm Antenna status | 540 | 67.5 | 33.75 | 16.875 | Radio |
| | RF I2C | I2CfailureRF | 1 | Unsigned | | 17 | - | RF I2C Bus failure indicator | 542 | 67.75 | 33.875 | 16.9375 | Computer Hardware |
| | NONE | pad1 | 1 | Unused | | 0 | - | Filler | 543 | 67.875 | 33.9375 | 16.96875 | NONE |
| | WOD Stored | wodSize | 8 | | | 1 | 000s | Number of WOD data payloads kept for each of Science and Housekeeping. In hundreds | 544 | 68 | 34 | 17 | Computer Software |
| | Diagnostic | swCmds | 32 | uint8_t | | 35 | - | ICR Diagnostic information | 552 | 69 | 34.5 | 17.25 | Command Receiver (ICR) |
| | HW Command Count | hwCmdCnt | 6 | uint8_t | | 1 | - | Number of hardware commands since last reset | 584 | 73 | 36.5 | 18.25 | Command Receiver (ICR) |
| | SW Command Count | swCmdCnt | 6 | uint8_t | | 1 | - | Number of software commands since last reset | 590 | 73.75 | 36.875 | 18.4375 | Command Receiver (ICR) |
| | NONE | pad2 | 28 | Unused | | 0 | - | Filler | 596 | 74.5 | 37.25 | 18.625 | NONE |
| **END** | | | | | | | | **Total Length** | 624 | 78 | 39 | 19.5 | |

Structure: maxSpecific_t, file: maxDownlink.h

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IHU Hard Error Data | IHUhardError | 32 | Unsigned | DownlinkError_t | 19 | - | Reason for last reset | 496 | 62 | 31 | 15.5 | Computer Hardware |
| | MAX Timestamp Reset Count | maxTimestampReset | 16 | Unsigned | | 1 | s | Timestamp of last max change | 528 | 66 | 33 | 16.5 | NONE |
| | MAX Timestamp Uptime | maxTimestampUptime | 25 | Unsigned | | 1 | s | Timestamp of last max change | 544 | 68 | 34 | 17 | NONE |
| | Safe Mode | SafeModeIndication | 1 | Unsigned | | 21 | - | In safe mode? | 569 | 71.125 | 35.5625 | 17.78125 | Computer Software |
| | Auto Safe Mode | AutoSafeModeActive | 1 | Unsigned | | 21 | - | In Autosafe mode? | 570 | 71.25 | 35.625 | 17.8125 | Computer Software |
| | Auto Safe Allowed | AutoSafeModeAllowed | 1 | Unsigned | | 21 | - | Autosafe enabled? | 571 | 71.375 | 35.6875 | 17.84375 | Computer Software |
| | Science Mode | ScienceModeActive | 1 | Unsigned | | 21 | - | In Science Mode? | 572 | 71.5 | 35.75 | 17.875 | Computer Software |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NONE | pad1 | 19 | Unused | | 0 | - | Make it even | 573 | 71.625 | 35.8125 | 17.90625 | NONE |
| | NONE | pad2 | 32 | Unused | | 0 | - | | 592 | 74 | 37 | 18.5 | NONE |
| END | | | | | | | | **Total Length** | 624 | 78 | 39 | 19.5 | |

Structure: minSpecific_t, file: minDownlink.h

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Soft Error | IHUsoftError | 32 | Unsigned | DownlinkSoftError | 20 | - | Diagnostic Data on IHU Software Errors | 496 | 62 | 31 | 15.5 | Computer Software |
| | NONE | minTimestampReset | 16 | Unsigned | | 1 | s | Timestamp Resets | 528 | 66 | 33 | 16.5 | NONE |
| | NONE | minTimestampUptime | 25 | Unsigned | | 1 | s | Timestamp uptime | 544 | 68 | 34 | 17 | NONE |
| | NONE | SafeModeIndication | 1 | Unsigned | | 21 | - | In safe mode? | 569 | 71.125 | 35.5625 | 17.78125 | NONE |
| | NONE | AutoSafeModeActive | 1 | Unsigned | | 21 | - | In Autosafe mode? | 570 | 71.25 | 35.625 | 17.8125 | NONE |
| | NONE | AutoSafeModeAllowed | 1 | Unsigned | | 21 | - | Autosafe enabled? | 571 | 71.375 | 35.6875 | 17.84375 | NONE |
| | Science Mode | ScienceModeActive | 1 | Unsigned | | 21 | - | In Science Mode? | 572 | 71.5 | 35.75 | 17.875 | NONE |
| | NONE | pad1 | 19 | Unused | | 0 | - | Make it even | 573 | 71.625 | 35.8125 | 17.90625 | NONE |
| | NONE | pad2 | 32 | Unused | | 0 | - | Make it even | 592 | 74 | 37 | 18.5 | NONE |
| END | | | | | | | | **Total Length** | 624 | 78 | 39 | 19.5 | |

Structure:radiation_t, file:radDownlink.h

| // | **Field Name** | **Code Name** | **Size (Bits)** | **Type** | **C Type** | **Conversion** | **Label** | **Description** | Bit Offset | Byte Offset | Short Offset | Long offset | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NONE | data | 464 | Unsigned | uint8_t | 0 | - | 0x01 specifies Fox-1A (each Fox satellite will have a unique ID) | 0 | 0 | 0 | 0 | NONE |
| | NONE | pad1 | 104 | | | 0 | - | Room for optional more science data | 464 | 58 | 29 | 14.5 | NONE |
| | NONE | pad2 | 6 | | | | | Even it up | 568 | 71 | 35.5 | 17.75 | NONE |
| | Exp1 Board Number | exp1BoardNum | 2 | Unsigned | | | | Experiment Board Active | 574 | 71.75 | 35.875 | 17.9375 | NONE |
| | WOD Timestamp Reset Count | WODTimestampReset | 16 | Unsigned | | 1 | - | Time Whole Orbit Data was collected (WOD only) | 576 | 72 | 36 | 18 | NONE |
| | MAX Timestamp Uptime | WODTimestampUptime | 25 | Unsigned | | 1 | s | Time Whole Orbit Data was collected (WOD only) | 592 | 74 | 37 | 18.5 | NONE |
| | CRC Error on this WOD entry | WodCRCError | 1 | Unsigned | | 0 | - | Memory error (WOD only) | 617 | 77.125 | 38.5625 | 19.28125 | NONE |
| | Pad1 | pad3 | 6 | | | | - | Make it even | 618 | 77.25 | 38.625 | 19.3125 | NONE |
| END | | | | | | | | | 624 | 78 | 39 | 19.5 | |

//
Structure:wodSpecific_t,file:wodSpecificDownlink.h

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IHU Hard Error Data | IHUhardError | 32 | Unsigned | DownlinkError_t | 19 | - | Reason for last reset | 496 | 62 | 31 | 15.5 | Computer Hardware |
| | Diagnostic Data | IHUdiagData | 32 | Unsigned | DownlinkIHUInfo | 18 | - | Diagnostic Data on IHU Performance | 528 | 66 | 33 | 16.5 | Computer Software |
| | WOD Timestamp Reset Count | WODTimestampReset | 16 | Unsigned | | 1 | - | Time Whole Orbit Data was collected | 560 | 70 | 35 | 17.5 | NONE |
| | MAX Timestamp Uptime | WODTimestampUptime | 25 | Unsigned | | 1 | s | Time Whole Orbit Data was collected | 576 | 72 | 36 | 18 | NONE |
| | Safe Mode | SafeModeActive | 1 | Unsigned | | 21 | - | Are we in safe mode? | 601 | 75.125 | 37.5625 | 18.78125 | Computer Software |
| | Auto Safe Mode | AutoSafeModeActive | 1 | Unsigned | | 21 | - | Are we in safe mode? | 602 | 75.25 | 37.625 | 18.8125 | Computer Software |
| | Exp 1 | expFailure0 | 1 | Unsigned | | 17 | - | Experiment 1 Failure Indicator | 603 | 75.375 | 37.6875 | 18.84375 | Experiments |
| | Exp 2 | expFailure1 | 1 | Unsigned | | 17 | - | Experiment 2 Failure Indicator | 604 | 75.5 | 37.75 | 18.875 | Experiments |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Exp 3 | expFailure2 | 1 | Unsigned | | 17 | - | Experiment 3 Failure Indicator | 605 | 75.625 | 37.8125 | 18.90625 | Experiments |
| | Science Mode | ScienceModeActive | 1 | Unsigned | | 17 | - | Are we in science mode? | 606 | 75.75 | 37.875 | 18.9375 | Computer Software |
| | Battery I2C | I2CfailureBatt | 1 | Unsigned | | 17 | - | Battery I2C Bus failure indicator | 607 | 75.875 | 37.9375 | 18.96875 | Computer Hardware |
| | PSU1 I2C | I2CfailurePSU1 | 1 | Unsigned | | 17 | - | PSU2 I2C Bus failure indicator | 608 | 76 | 38 | 19 | Computer Hardware |
| | PSU2 I2C | I2CfailurePSU2 | 1 | Unsigned | | 17 | - | PSU2 I2C Bus failure indicator | 609 | 76.125 | 38.0625 | 19.03125 | Computer Hardware |
| | RF I2C | I2CfailureRF | 1 | Unsigned | | 17 | - | RF I2C Bus failure indicator | 610 | 76.25 | 38.125 | 19.0625 | Computer Hardware |
| | CRC Error | WodCRCError | 1 | Unsigned | | 0 | - | CRC Error on this WOD entry | 611 | 76.375 | 38.1875 | 19.09375 | Computer Software |
| | HW Command Count | hwCmdCnt | 6 | uint8_t | | 1 | - | Number of hardware commands since last reset | 612 | 76.5 | 38.25 | 19.125 | Command Receiver (ICR) |
| | SW Command Count | swCmdCnt | 6 | uint8_t | | 1 | - | Number of software commands since last reset | 618 | 77.25 | 38.625 | 19.3125 | Command Receiver (ICR) |
| END | | | | | | | | Totals: | 624 | 78 | 39 | 19.5 | |

**Date:** September 12, 2017
**Version:** Version 1.02a

# AMSAT *Fox-1E (RadFxSat-2)*

# IHU to Experiment 1 Interface Control Document

## 1 Introduction

This document specifies the message interface between the Internal Housekeeping Unit (IHU) and the Experiment System in Position 1 of the satellite, known as the Vanderbilt University Phoenix Payload and abbreviated herein as EXP1.

### 1.1 Document History

| DATE | VERSION | SUMMARY |
|---|---|---|
| December 15, 2015 | 1.00 | Initial version |
| December 16, 2015 | 1.01 | Change Vulcan to Phoenix |
| June 21, 2017 | 1.02 | Changes for RadFxSat-2 |
| September 12, 2017 | 1.02a | Add s/w version and experiment period units. |

## 1.2 Document Scope

This document will specify the control of EXP1, the messaging format, and the I²C bus hardware operation for the communications between the IHU and the EXP1.

## 1.3 References

1. AMSAT *Fox-1,* System Requirements Specification
2. AMSAT *Fox-1*, System Design Specification
3. AMSAT *Fox-1*, IHU Software Architecture Specification
4. Vanderbilt University Phoenix Payload Interface Control Document

# 2 General Messaging Requirements

## 2.1 Link Protocol Requirements

2.1.1 The IHU shall initiate a command to the EXP1.

2.1.2 The EXP1 shall send a reply to each IHU request.

2.1.3 Message bit order shall be Big Endian.

2.1.4 The IHU shall determine the action to be taken in the event of an invalid, improper, or missing message from the EXP1.

2.1.5 The EXP1 shall take no action in the event of an invalid or improper message from the IHU.

2.1.6 Message byte order shall be Big Endian.

## 2.2 General Message Requirements

2.2.1 Each message shall contain a header block.

2.2.2 Each message shall contain a packet error check (PEC) in the form of CRC8.

2.2.2.1 The message address byte shall be included when calculating the CRC8.

## 2.3 I²C 1 Bus Hardware Interface Requirements

2.3.1 The I²C Vdd shall be 3.0V.

2.3.2 The bus speed shall be Fast (400kbit/s).

2.3.3 The EXP1 I²C 7 bit address shall be 0x2A.

# 3 Experiment Operation

## 3.1 Experiment Power Control

3.1.1 The IHU shall exert control over the power state of the EXP1 by the Experiment Enable 1 pin on the satellite bus as shown in Table 1.

**Table 1**

| Pin State | Description |
|---|---|
| High | Power On Experiment |
| Low or high-impedance | Power Off Experiment |

3.1.2 The IHU shall not power on the experiment if the power bus voltage (VBATT) is less than or equal to 3.3 Volts.

3.1.3 The IHU shall perform the Experiment Cease Operation Sequence and the Experiment Power Off Sequence if the power bus voltage (VBATT) falls to less than or equal to 3.3 Volts while the experiment is powered on.

## 3.2 Experiment Power On Sequence

3.2.1 The IHU shall set and hold the Experiment Enable 1 pin HIGH.

3.2.2 The IHU shall not send any message to the EXP1 for a minimum of 100 milliseconds.

3.2.3 The IHU shall send a Set Time command to the EXP1.

## 3.3 Experiment Begin Operation Sequence

3.3.1 Upon completion of the Power On sequence the IHU shall send a Set Run State Active command message to the EXP1.

## 3.4 Experiment Cease Operation Sequence

3.4.1 The IHU shall send a Set Run State Halt command message to the EXP1.

3.4.2 The IHU shall not send any message to the EXP1 for a minimum of 10000 milliseconds.

3.4.3 The IHU shall send a Set Run State Standby command message to the EXP1.

## 3.5 Experiment Power Off Sequence

3.5.1 The IHU shall set the Experiment Enable 1 pin LOW.

3.5.1.1 The absence of a HIGH state on the Experiment Enable 1 pin shall be construed as a LOW state whether the pin is actually LOW, or in a high-impedance state.

# 4 Message Content Requirements

## 4.1 Command Message

4.1.1 The message header block shall be constructed as shown in table 2.

4.1.2 The message header block shall be sent with each Command and Response block.

**Table 2**

| Field | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| Message Version | 2 | Unsigned | 0x01 | 0xFFFF | Message ICD version |
| Software Build | 2 | Unsigned | 0x01 | 0xFFFF | Software Build version |

4.1.2.1 The Message Version shall be an integer representing the IHU to EXP1 ICD document version number from which the message format is derived, having the decimal point removed (e.g. version 1.03 would be 103 decimal or 0x67). For this ICD version the message version is 0x66, which is 102.

4.1.2.2 The Software Build shall be an integer representing the software build version number of the system originating the message, having any decimal points removed (e.g. version 1.03 would be 103 decimal or 0x67). In practice, this is value is not changed. The current value used is 100.

## 4.2 Command Message Block

4.2.1 The command message block shall be constructed as shown in Table 3.

**Table 3**

| Field | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| COMMAND | 2 | Unsigned | 0x0000 | 0x8202 | Hexadecimal Command |
| ARGUMENT | Variable | Unsigned | - | - | Optional Arguments As Required |

The command message block shall contain one command in the COMMAND field as shown in Table 4.

Table 4

| Command Name | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| Nop | 2 | Unsigned | 0x0000 | 0x0000 | No effect; response undefined. Test for I²C acknowledgement only. |
| Echo | 2 | Unsigned | 0x0001 | 0x0001 | Echo this byte stream |
| Resend | 2 | Unsigned | 0x0002 | 0x0002 | Resend last result |
| Get UID | 2 | Unsigned | 0x0003 | 0x0003 | Controller 7 byte identifier |
| Get Status | 2 | Unsigned | 0x0004 | 0x0004 | Controller status indication |
| Get Diagnostics | 2 | Unsigned | 0x0006 | 0x0006 | Self-check Diagnostic |
| Get Telemetry | 2 | Unsigned | 0x0010 | 0x0010 | Send telemetry data |
| Set Run State | 2 | Unsigned | 0x0080 | 0x0080 | Enter specified Run State |
| Get Run State | 2 | Unsigned | 0x0081 | 0x0081 | Query current Run State |
| Set Time | 2 | Unsigned | 0x0100 | 0x0100 | Number of seconds since epoch |
| Get Time | 2 | Unsigned | 0x0101 | 0x0101 | Number of seconds since epoch |
| Get Data | 2 | Unsigned | 0x0280 | 0x0280 | Send (number of bytes) data |
| Set Experiment Periods | 2 | Unsigned | 0x8200 | 0x8200 | Set active time in minutes for each experiment |
| Set Current Experiment | 2 | Unsigned | 0x8201 | 0x8201 | Force an experiment to be active |
| Get Exp | 2 | Unsigned | 0x8202 | 0x8202 | Get Current Experiment |

4.2.3 The command message shall contain arguments for the Echo command, as shown in Table 5.

**Table 5**

| Field | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| ARGUMENT | 4 | Unsigned | - | - | Data to be echoed |

4.2.4 The command message shall contain one argument for the Set Run State command, as shown in Table 6.

**Table 6**

| Run State | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| STANDBY | 2 | Unsigned | 0x0001 | 0x0001 | Enter Standby State |
| ACTIVE | 2 | Unsigned | 0x0003 | 0x0003 | Activate Experiments |
| HALT | 2 | Unsigned | 0x0005 | 0x0005 | Terminate Experiments |

4.2.5 The command message shall contain arguments for the Set Time command, as shown in Table 7.

**Table 7**

| Argument | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| IHU Reset Counter | 16 | Unsigned | 0x00 | - | Count of the number of IHU resets from non-volatile FRAM |
| MET Timestamp | 32 | Unsigned | - | - | MET timestamp (seconds since last IHU reset) |

4.2.6 The command message shall contain arguments for the Get Data command, as shown in Table 8.

**Table 8**

| Argument | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| BYTES TO SEND | 2 | Unsigned | 0x00 | 0x00FF | Number of bytes to send (1-256) |

4.2.7 The Set Experiment message shall contain an argument as shown in table 9

**Table 9**

| Argument | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| ExpNum | 2 | Unsigned | 0x0001 | 0x0004 | Experiment Number |

4.2.8 The Set Experiment Periods message shall contain four arguments as shown in table 9

**Table 10**

| Argument | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| Exp1 Time | 2 | Unsigned | 0 | 2880 | Time in minutes spent in experiment 1 |
| Exp2 Time | 2 | Unsigned | 0 | 2880 | Time in minutes spent in experiment 2 |
| Exp3 Time | 2 | Unsigned | 0 | 2880 | Time in minutes spent in experiment 3 |
| Exp4 Time | 2 | Unsigned | 0 | 2880 | Time in minutes spent in experiment 4 |

If a non-existent or disabled experiment is specified, when that experiment's turn comes, nothing will run for the specified time.

## 4.3 Response Message Block

4.3.1 The response message block shall be constructed as shown in Table 9.

**Table 11**

| Field | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| RESERVED | 1 | Unsigned | - | - | Reserved, ignore |
| ERROR CODE | 1 | Unsigned | 0x0000 | 0x0006 | Response to Command |
| LENGTH | 2 | Unsigned | 0x00 | 0xFFFF | Length of Return Value in Bytes |
| RETURN VALUE | Variable | Variable | - | - | Return Value |

4.3.2 The Error Code shall contain one code as shown in table 10.

**Table 12**

| Name | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| CMD_OK | 1 | Unsigned | 0x0000 | 0x0000 | Command invoked successfully |
| CMD_OP_ERR | 1 | Unsigned | 0x0001 | 0x0001 | Command not recognized |
| CMD_FORMAT_ERR | 1 | Unsigned | 0x0002 | 0x0002 | Incorrect command argument length |
| CMD_RANGE_ERR | 1 | Unsigned | 0x0003 | 0x0003 | Argument(s) out of bounds |
| CMD_PEC_ERR | 1 | Unsigned | 0x0004 | 0x0004 | Error check (CRC) mismatch |
| CMD_EXEC_ERR | 1 | Unsigned | 0x0005 | 0x0005 | Execution error |
| CMD_VERSION_ERR | 1 | Unsigned | 0x0006 | 0X0006 | Header Message Version mismatch |

4.3.3 The Status Flags for a GET STATUS response message shall be represented as individual bit values of a 16 bit RETURN VALUE as shown in Table 11.

**Table 13**

| Name | Bit Number | Description |
|---|---|---|
| REBOOTED | 0 | 1 = Experiment has rebooted – **NOT USED** |
| DATA READY | 1 | 1 = Experiment data available |
| TIME REQUEST | 2 | 1 = Request SET TIME |
| FAILED RUN STATE | 3 | 1 = Failed the run state – **NOT USED** |
| COMPLETED RUN STATE | 4 | 1 = Completed the run state – **NOT USED** |
| RESERVED | 5-15 | Always 0 |

4.3.4 The response message to a Set Time command shall contain one of the values as shown in Table 12.

**Table 14**

| Response Name | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| SUCCESS | 2 | Signed | 0x00 | 0x00 | Time Set successfully |
| FAILURE | 2 | Signed | 0xFFFF | 0xFFFF | Time Set failed |

4.3.1 The response message to a Get Experiment command shall contain one of the values as shown in Table 12. Note: The value 0 indicates there are no experiments active.

**Table 15**

| Response Name | Size (Bytes) | Type | Min Value | Max Value | Description |
|---|---|---|---|---|---|
| Value | 2 | Unsigned | 0 | 4 | Current experiment |

# 5 Message Integrity

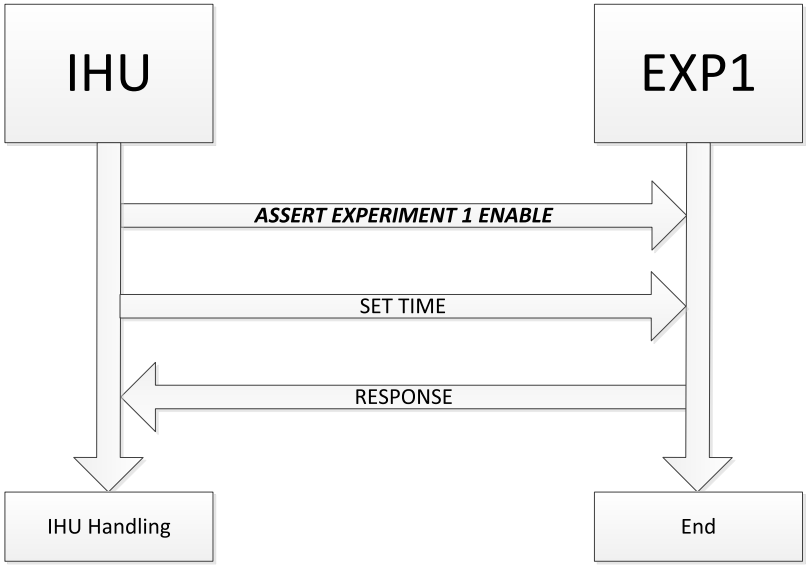
## 5.1 Invalid Messages

5.1.1 If the PEC (CRC8) fails, the message shall be considered invalid.

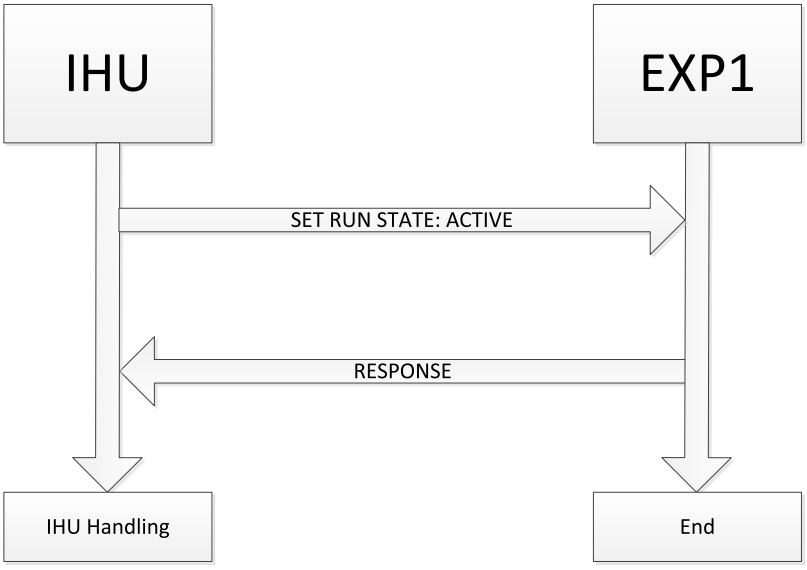
5.1.2 If the Message Version does not match the message version in use for the construction of messages on the receiving system, the message shall be considered invalid.

# 6 Message Flow Diagrams

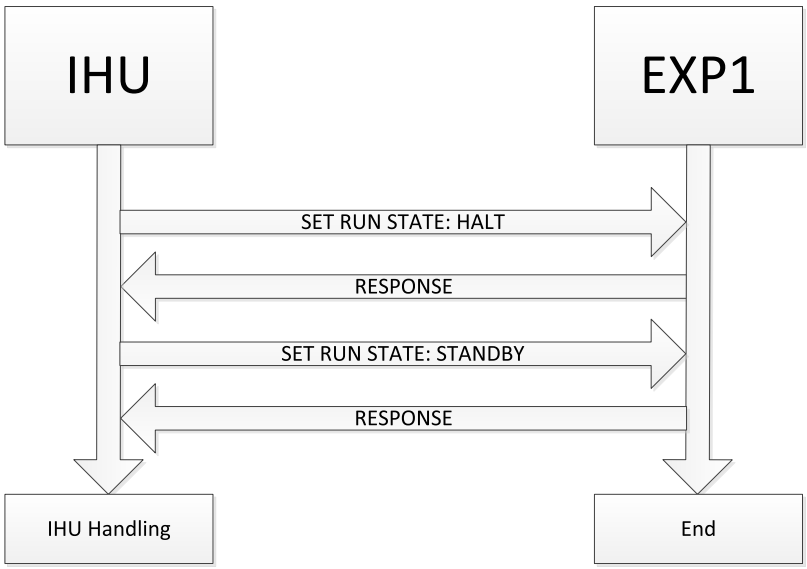
## 6.1 EXPERIMENT POWER ON SEQUENCE



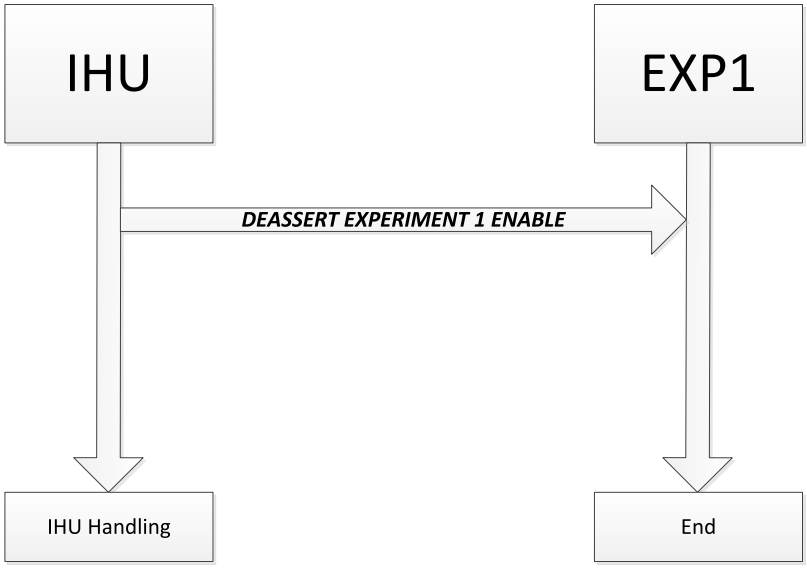

## 6.2 EXPERIMENT BEGIN OPERATION SEQUENCE



## 6.3 EXPERIMENT CEASE OPERATION SEQUENCE




## 6.4 EXPERIMENT POWER OFF SEQUENCE



## 6.5 SERVICING EXPERIMENT OPERATION

**Figure 1**

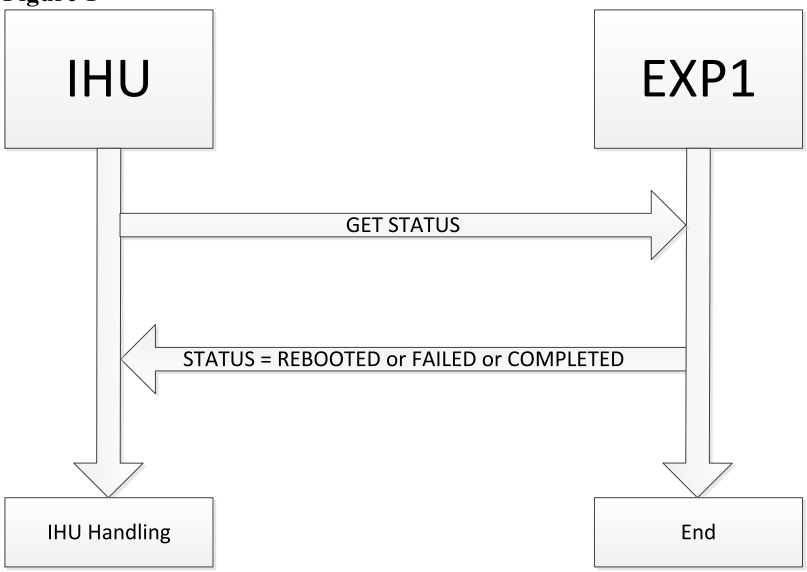



**Figure 2**

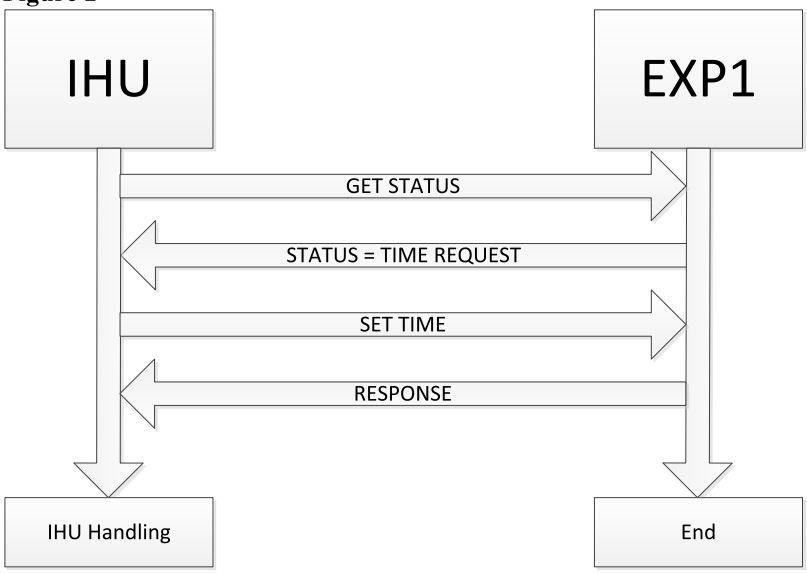
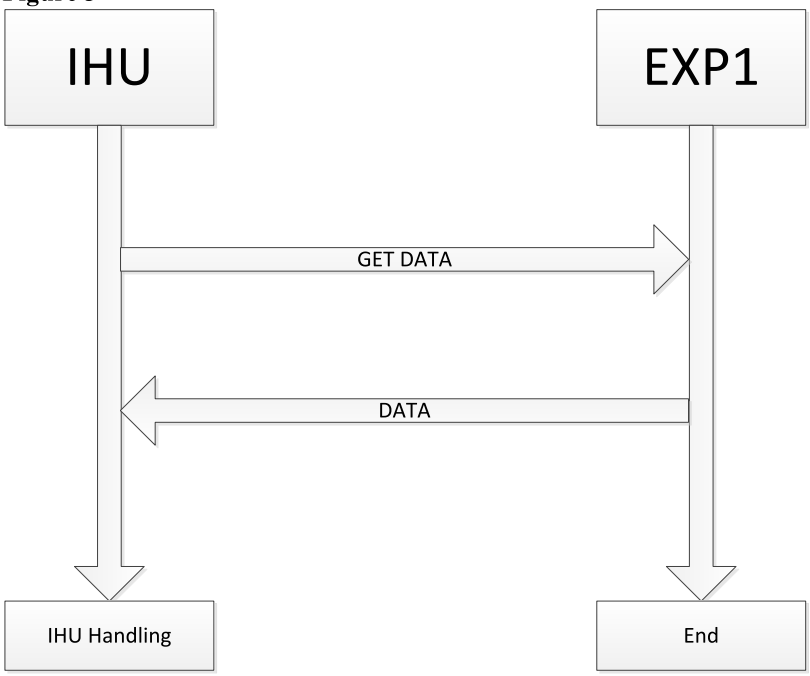


**Figure 3**



# RadFXSat-2 / Fox IHU Interface Control Document (ICD)

**Date: January 3, 2017**
**Revision: v1.0**

## ICD Revision History

| Revision | Date | Author | Change Log |
|---|---|---|---|
| v0.0 | August 30, 2016 | E. Skoog (K1TVV) | ICD draft template |
| v1.0 | January 3, 2017 | B. Fisher (W2BFJ); J. Klingelhoeffer (WB4LNM); Chris Thompson (AC2CZ); | Initial proposed IHU-LTI Control, Experiment and Spacecraft Telemetry Line Assignments |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

# 1. INTRODUCTION

Interface management and control is a System Engineering activity that begins in parallel with the development of system architectures and continues for the life of a system. Likewise, the evolutionary process to define interfaces begins with concept definition and continues throughout the development and design phases. Interfaces, both internal and external to the system, are documented and formally managed. This activity is intended to ensure compatibility among subsystems being designed and fabricated, and to ensure future interoperability between systems.

System interfaces are both physical and functional. Initially, a system is depicted by a top-level Concept Diagram that identifies its external interfacing elements. Source documents, such as Capability Development Documents, Capstone Requirements Documents, external system specifications, and interface documents provide interface requirements to ensure interoperability between systems and that the required capabilities are realized. An operational concept may also provide descriptions, interactions, and requirements between a system and its external elements. An interface definition process will evolve interface architectures and requirements in conjunction with system definition, i.e., interfaces will mature as the operational and system requirements mature.

## 1.1. Overview

An interface, from the perspective of system development, can be identified as any point where a system and something, or someone, meet. This interface point may include other systems, internal hardware, external peripherals, networks, system users, etc.

Interfaces are documented using Interface Control Documents (ICD) that describe the system's interfaces including any rules for communicating with them. ICDs help ensure compatibility between system segments and components and are considered key elements of effective system design and development.

**Interface Control Document (ICD)** ≡ describes the interworking of two elements of a system that share a common interface. An ICD may also describe the interaction between a user and a system, a software component and a hardware device, or two software components. An ICD is typically used where complex interfaces exist between components that are being developed by different teams. It is jointly prepared by the interfacing groups.

ICDs define for the development team applicable system inputs, outputs, internal interfaces, and interfaces to other systems, subsystems, and/or users. An ICD should only describe the interface itself and not the characteristics of the systems connected. An ICD should also not include details regarding the meaning or intended use of the data being interchanged across the interface. Any design features, functions, or logic supporting a system interface should be detailed in separate design and/or specification documents. Defining an ICD in this way allows other teams to develop connecting systems without concern of design details or how data is treated by the other system. This allows development teams to work without knowledge of the logic or technical aspects behind the interfacing system. This facilitates modularity and leads to easier maintenance and extensibility of systems.

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

## 1.2. Purpose

The purpose of interface management and control is to define interface requirements so as to ensure compatibility between inter-connected pieces of equipment and to provide an authoritative means of controlling the design of the interfaces.

The purpose of an ICD is to clearly communicate a system's external inputs and outputs for all potential interface actions whether or not they are transparent to the system's users. An ICD may describe the system interface to the lowest physical elements and parameter values (e.g., circuits, voltages, impedances).

An ICD describes the relationship between two components of a system in terms of data items and messages passed, protocols used, and the timing and sequencing of interaction events. Specifically, ICDs are generated to:

- Ensure configuration control of the system's interface design to prevent any changes to characteristics that would affect compatibility with other systems,
- Define and illustrate the physical and functional characteristics of a system in sufficient detail to ensure compatibility of the interface, and in order that compatibility can be determined from the information in the ICD alone,
- Facilitate identification of missing interface data and control the submission of that data, and
- Communicate coordinated design decisions and design changes to all involved parties.

ICD's by their nature are requirements documents. They define design requirements and facilitate integration. They record the agreed-to design solution to interface requirements and provide a control mechanism to ensure the agreed-to design is not changed by one participant without agreement with the other participant(s).

To be effective, ICD's should follow a schedule compatible with the design maturation of the system, e.g., initial ICD's should be at the 80% level of detail at Preliminary Design Review (PDR), should mature as the design matures, and should reach the 99% mark near the Critical Design Review (CDR) period.

## 1.3. Scope

This document details the signal line interface design between the Fox-1E's Integrated Housekeeping Unit (IHU) and the LT cards on Fox-1E and future LT deployments.

Since the IHU and several other boards for Fox-1E have already been designed and built according to bus specifications from previous Fox-1 satellites, the only bus lines available are either point-to-point signals previously used on the FM Foxes to connect the TX or RX boards with the IHU, or bus lines to which multiple boards could connect.

The Fox-1E's basic LT design consists of two interconnected boards: a Receiver/Transmitter (RX/TX) board and an Improved Command Receiver (ICR) board. The ICR board includes an I2C multiplexer chip which can accommodate up to 8 analog channel inputs for Analog to Digital conversion and multiplexing of sensor data for efficient transmission to the IHU without tying up additional Fox Bus lines. The LT boards are monitored and controlled by the IHU.

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

## 1.4. System Identification

AMSAT-NA's Fox-1E (RadFXSat-2) CubeSat will feature the first linear transponding capability of the Fox-1x family of satellites. The Linear Transponder will provide simultaneous multi-channel communications using primarily CW and SSB. The design of the transponder will include state-of-the-art integrated circuit components to provide exceptional communications performance, while minimizing size, weight and power. Additionally, unlike previous Fox-1x CubeSats, Fox-1E will be a V/U satellite. This means that operator uplink transmissions will require generation of VHF (2 meter) SSB/CW amateur radio signals and downlink signal receipt will require reception of UHF (70 cm) SSB/CW signals. A separate 1200 bps BPSK modulated channel will be dedicated to downlinking CubeSat telemetry and experiment data.

BLOCK DIAGRAM - At the conceptual level, the Fox-1E Linear Transponder can be envisioned as a set of major subsystem functional blocks as illustrated in Figure 1 below.

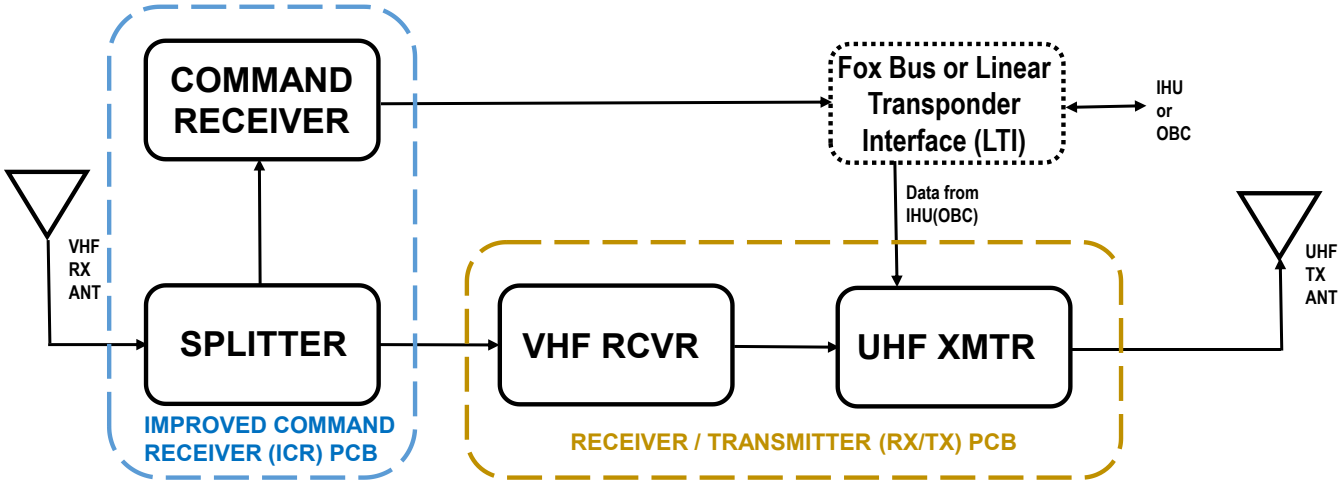



**Figure 1: Fox-1E Linear Transponder Top Level Block Diagram**

The Linear Transponder RF development team's design goal is to complete the design on two Printed Circuit Boards (PCB). However, interfacing an LT with non-IHU equipped spacecraft, will require development of a third (unique) Linear Transponder Interface (LTI) PCB.

FREQUENCIES - The Linear Transponder shall exhibit a transmit/receive (3dB) bandwidth of 30kHz. The currently planned uplink-downlink frequencies (subject to IARU coordination and FCC authorization) for on-orbit operations are:

**UPLINK**: 145.860 MHz to 145.890 MHz

**DOWNLINK**: 435.790 MHz to 435.760 MHz (inverting)

**TLM Downlink Channel** (dedicated): 435.750MHz (1200 bps BPSK)

**COMMAND & CONTROL UPLINK CHANNEL**: "anywhere" in the 145.8 to 146.0MHz International Amateur Satellite Service frequency band.

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

# 2. INTERFACE DEFINITIONS

## 2.1. Fox Bus Signal Lines

TABLE 1 is a list of the Fox Bus signal lines involved in the operation of the Linear Transponder. Lines capable of handling analog signals are shaded in yellow.

| Pin | Line Name | Source | Sink | Use |
|---|---|---|---|---|
| 2 | TX_PA_ CURRENT | RX/TX | IHU | Power Amplifier current consumption; analog signal; 0 to 3V |
| 3 | SPI1_SCLK | IHU | RX/TX | SPI Bus Slave Clock for TLM data modulator IC |
| 4 | TX_TEMP | RX/TX | IHU | Power Amplifier temperature; analog signal; 0 to 3V |
| 5 | SPI1_MISO | RX/TX | IHU | SPI bus Master In, Slave Out data from TLM modulator IC |
| 7 | SPI1_MOSI | IHU | RX/TX | SPI bus Master Out, Slave In data to the TLM modulator IC |
| 8 | VGA_CNTL | RX/TX | IHU | Control voltage to the receive chain's Variable Gain Amplifier; analog signal; 0 to 3V |
| 10 | ICR_RSSI | ICR | IHU | Analog Received Signal Strength Indication from the command receiver; analog signal; 0 to 3V |
| 12 | TLM_DATA | IHU | RX/TX | Telemetry and/or Experiment Data to be transmitted on the TLM downlink channel (+3V = Mark, 0V = Space) |
| 16 | CMD_H/W_Bit_0 | ICR | IHU | Hardware Command Decoder Bit 0 (LSB) |
| 18 | CMD_H/W_Bit_1 | ICR | IHU | Hardware Command Decoder Bit 1 |
| 20 | CMD_H/W_Bit_2 | ICR | IHU | Hardware Command Decoder Bit 2 (HIGH = IHU OFF) |
| 22 | CMD_H/W_Bit_3 | ICR | IHU & RX/TX | Hardware Command Decoder Bit 3 (HIGH = INHIBIT TRANSMIT) |
| 24 | CMD_H/W_BITS_STROBE | ICR | IHU | HIGH = Command Data Valid |
| 26 | RX/TX_CNTL | IHU | RX/TX | HIGH = IHU is running and in control<br>LOW = LT functions "Autonomously" |
| 28 | SPI1_SS | IHU | RX/TX | SPI1 Slave Select to TLM modulator IC |
| 30 | PB_ENABLE | IHU | RX/TX | LT User Passband Transmission Enable |
| 32 | TX_PWR_FWD | RX/TX | IHU | Power amplifier (FORWARD) Transmit Power; analog signal; 0 to 3V |
| 34 | CMD_SERIAL_DATA | ICR | IHU | Command Receiver Serial Data Stream (+3V = Mark, 0V = Space) |
| 37 | I2C2_SCL | IHU | MPPT(2)<br>BATT(1)<br>LT(1) | I²C Standard Bus CLOCK (Bidirectional):, IHU master and four slaves (2 MPPTs, 1 BATT, and 1 LT) |
| 38 | SENSOR_PWR | IHU | MPPT(2)<br>BATT(1)<br>LT(1) | IHU generated and latch-up protected +3 volts for telemetry sensors and all I2C Multiplexer ICs |
| 39 | I2C2_SDA | IHU | MPPT(2)<br>BATT(1)<br>LT(1) | I²C Standard Bus DATA (Bidirectional):, IHU master and four Slaves (2 MPPTs, 1 BATT, and 1 LT) |
| 40 | TLM_ENABLE | IHU | RX/TX | Enable telemetry and/or experiment data transmission on the Telemetry Downlink channel |
| 42 | MODEM_CXR_DET | ICR | IHU | ICR modem carrier detect. High if data is detected on the command uplink. |
| 52 | TX_PWR_RFLT | RX/TX | ICR | Power amplifier (REFLECTED) Transmit Power; analog signal; 0 to 3V carried by Fox Bus to ICR card's ADC/MUX |

**TABLE 1: LT - IHU INTERCONNECT LINES**

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

Additionally, the following Fox Bus Signal Pins are available for use between the LT cards if needed. They were not used on any previous Fox satellites. Note these lines pass through all cards on the bus, therefore they are not suitable for RF signals.

FOX Bus Pins: 14, 21, 27, 36, 54

## 2.2. Linear Transponder RX/TX Card CONTROL Lines

The LT's RX/TX card is normally[1] controlled by the Integrated Housekeeping Unit (IHU). Seven (7) Fox Bus control lines originate from the IHU and connect to the RX/TX card. These control lines consist of two types: (a) multi-value commands, and (b) binary commands.

### 2.2.1 Multi-Value Commands

The LT's Telemetry Data BPSK Modulator (an LTC 5599 IC) is controlled over a Serial Peripheral Interface (SPI) bus. SPI is a full-duplex, synchronous serial communication interface using a master-slave architecture with a single bus master. In the Fox-1E LT application, the master is the IHU and the slave is the RX/TX card's LTC5599 IC. The Fox-1E's IHU controls the LTC 5599 IC over three Fox Bus lines: SPI1_SCLK (Bus Pin 3), SPI1_SS (Bus Pin 28), and SPI1_MOSI (Bus Pin 7). These control lines allow initialization and read/write actions for the modulator IC's registers.

The fourth multi-value command line is the TLM_DATA line (Bus Pin 12). This line provides the RX/TX card with fully encoded data strings representing spacecraft health and status telemetry data and/or data generated by the on-board, hosted experiment(s). Since this string data contains (pulse-shaped) binary bits, this line is considered the other 'multi-value' command line to be utilized, but only when the Telemetry downlink channel is enabled.

### 2.2.2 Binary Commands

The remaining three (3) RX/TX card control lines from the IHU carry simple binary commands, i.e., ON-OFF; ENABLE/DISABLE; HIGH/LOW; etc. Table 2 below details the logic states for these binary (HIGH/LOW) command bus signals that control the Linear Transponder's Receive/Transmit (RX/TX) card. In this table, the possible control signal states are shown on the left with no highlighted background. The resultant (operational) states of the LT's RX/TX card are shown on the right with green highlighted background. NOTE: the LT's operational state resulting from these lines is also enabled by one Command Hardware line, CMD_H/W_BIT_3, from the ICR as also shown in TABLE 2.

[1] If the IHU is inoperable, the Fox-1E's LT reverts to a 'degraded mode' state where it operates "autonomously." This operational state results when an inoperable IHU is unable to hold Fox Bus Pin 26 in a HIGH state.

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

| CMD-H/W_BIT_3 (Pin 22) | RX/TX_CNTL (Pin 26) | TLM_ ENABLE (Pin 40) | PB_ENABLE (Pin 30) | Result: Telemetry Operational State | Result: Transponder Operational State |
|---|---|---|---|---|---|
| HIGH | DON'T CARE | DON'T CARE | DON'T CARE | DISABLED[2] | DISABLED[2] |
| LOW | HIGH | HIGH | HIGH | ENABLED | ENABLED |
| LOW | HIGH | LOW | HIGH | DISABLED | ENABLED |
| LOW | HIGH | HIGH | LOW | ENABLED | DISABLED |
| LOW | HIGH | LOW | LOW | DISABLED[2] | DISABLED[2] |
| LOW | LOW | DON'T CARE | DON'T CARE | DISABLED | ENABLED (Autonomous Mode)[3] |

**TABLE 2: RX/TX Card Binary Command States**

### 2.3. Linear Transponder RX/TX Card MONITOR Lines

The LT's RX/TX card provides five (5) monitoring lines to the IHU. These lines carry signals generated from sensors on the RX/TX card representing health and status information to the IHU for processing and downlink telemetry reporting. The RX/TX card monitoring lines are carried on Fox Bus pins 2, 4, 5, 8, and 32 as detailed in TABLE 1.

### 2.4. Linear Transponder Improved Command Receiver (ICR) Card CONTROL Lines

To ensure RELIABLE control of the spacecraft under adverse conditions and degraded operational scenarios, the LT's ICR card is designed to function without requiring control direction from an IHU. Consequently, the control lines to the ICR are used to effect the transfer of LT collected sensor data ONLY when the IHU is functional. This is accomplished by aggregating sensor data via a multiplexer IC and communicating the resultant telemetry information to an IHU over an Inter-Integrated Circuit (I2C) bus, a multi-master, multi-slave, single-ended, serial computer bus.

The Fox-1E's IHU supports two (2) IC2 busses. The first bus, I2C1, is dedicated to hosted experiments. While the second, I2C2, is dedicated to Fox-1E health and status data monitoring. Each I2C bus supports multiplexed data reception and transmission control of four (4) connected ADC/Multiplexers. Specifically, I2C2 bus operations involve the Fox-1E's two Maximum Power Point Trackers (MPPTs), the Battery card, and the ICR. The I2C2 bus control signal lines are bidirectional, viz., I2C2_SCL slave clock (Pin 37) and I2C2_SDA slave data (Pin 39).

[2] In the case of a purposeful TX shut-down, either by IHU flight program software execution, or an ICR hardware bit 3 command, the TX/RX card will be shut-down/powered off, BUT the ICR is NEVER powered down/turned off!

[3] Similar to the Carrier Operated Relay (COR) mode of the FM Fox spacecraft, if the Fox-1E's IHU is inoperable AND Command Hardware Bit 3 is LOW, then the LT enters "Autonomous Mode" where user passband transponding is enabled, i.e., retransmission of uplink signals is supported, but no telemetry or experiment data is transmitted on the downlink.

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

Additionally, the IHU provides dedicated/conditioned power to the Fox-1E's analog sensors and all I2C Muxes over a SENSOR_PWR line (Fox Bus Pin 38). The IHU can manipulate this line to power cycle all of the IC2 Muxes. Consequently, this line can be considered a third control line to the ICR. These ICR control signal lines are detailed in TABLE 1.

## 2.5. Linear Transponder Improved Command Receiver (ICR) Card MONITOR Lines

### 2.5.1 Multiplexed Lines

The ICR card provides multiplexed telemetry information over an Inter-Integrated Circuit bus, I2C2. One of the four ADC/Multiplixers controlled on I2C2 is the ICR's ADC/MUX, a Texas Instrument ADS7828, a 12-Bit, 8-channel sampling ADC. This multiplexer facilitates increasing the number of LT monitored analog telemetry lines by converting such signals to digital streams, multiplexing, and transmitting them to the IHU via the shared I2C2 bus. The current allocation of the ICR MUX's eight (8) channels is detailed in TABLE 3 below:

| Multiplexer Channel | Telemetry Signal Name | Source Card | Monitored Information |
|---|---|---|---|
| 0 | TLM_PWR_RFLT | RX/TX | Linear Transponder's Power Amplifier (REFLECTED) Transmit Power; 0 to 3V analog signal supplied by Fox Bus Line #52 |
| 1 | +3V_REG | ICR | ICR card generated +3V regulated voltage |
| 2 | +3V_REG_PROT | ICR | ICR card generated +3V regulated voltage protected by a Load Management IC (FPF2004) |
| 3 | +2.5V_REG | ICR | ICR card generated +2.5V regulated voltage |
| 4 | +2.5V_PROT | ICR | ICR card generated +2.5V regulated voltage protected by a Load Management IC (FPF2004) |
| 5 | -------- | ------ | Spare Channel |
| 6 | -------- | ------ | Spare Channel |
| 7 | MUX_TEST | ICR | IHU-provided/controlled +3V SENSOR_PWR voltage reduced to 2.25V by a precision resistor voltage divider for sensor voltage monitoring and MUX conversion testing |

**TABLE 3: ICR Telemetry Multiplexer Channel Assignments**

### 2.5.2 Single Command Lines

In addition to sourcing (multiplexed) telemetry information over the I2C2 bus, the LT's ICR card provides eight (8) monitoring lines to the IHU. Four of these lines, CMD_H/W_Bit_0 through 3, might be more correctly categorized as 'control' lines as they govern critical spacecraft actions INDEPENDENT of the IHU as noted earlier. However, through Fox Bus lines (pins 16, 18, 20, and 22), these bit lines (strobed by pin 24) are made available to the IHU which can monitor their status as dictated by flight software. In addition to these four decoded Command Hardware Bits, the ICR card also provides the uplink tele-command serial data stream to the IHU through a CMD_SERIAL_DATA monitor line on Fox Bus pin 34. The remaining two ICR card monitoring lines are carried on Fox Bus pins 10 and 42. The usage of all these ICR Monitor lines is detailed in TABLE 1.

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

### 2.6. Fox-1E Linear Transponder Fox Bus Signal Line Diagram

Figure 2 below shows all LT signal lines as they are currently being implemented.

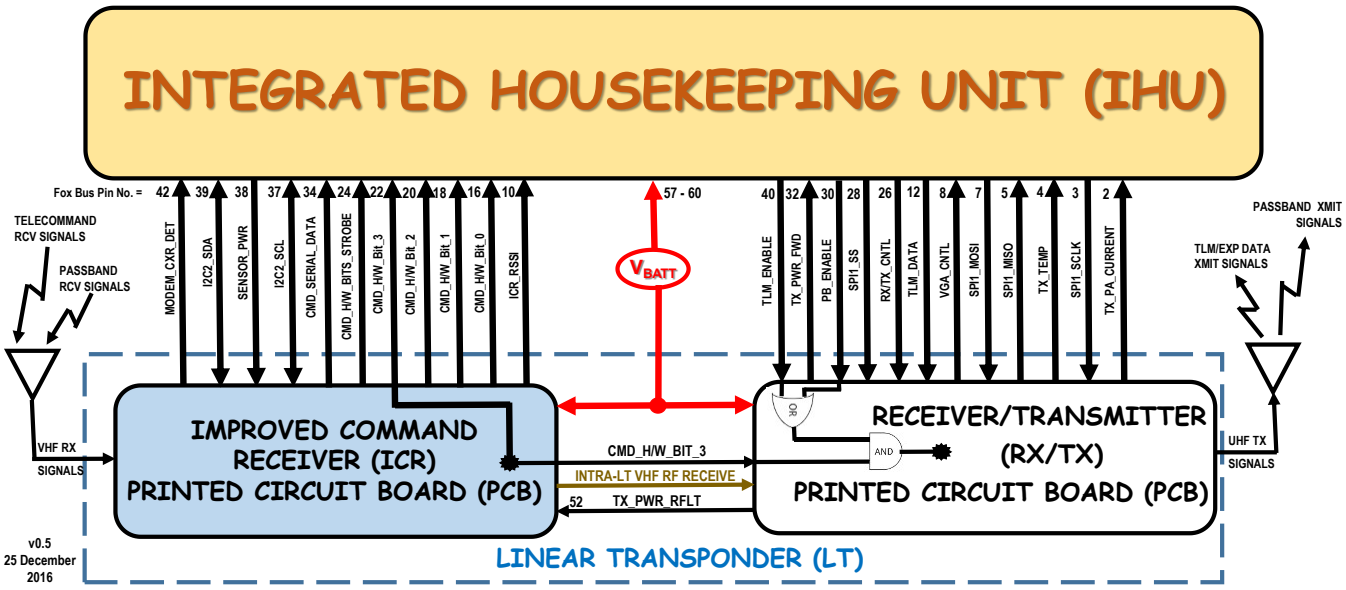



**FIGURE 2: Linear Transponder (LT) Fox Bus Signal Line Connections**

## 3. INTERFACE REQUIREMENTS

### 3.1. Interface Processing Time Requirements

### 3.2. Message/File Requirements

### 3.3. Communication Methods

### 3.4. Security Requirements

### 3.5. Thermal Requirements

### 3.6. Signal Requirements

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

# 4. INTERFACE CONTROL

## 4.1. Electrical Interface

## 4.2. Mechanical Interface

## 4.3. Data Interface

# 5. INTERFACE VERIFICATION

This section defines a set of qualification methods to be used to verify that the requirements for the interfaces defined in Section 3 have been met. Qualification methods include:

DEMONSTRATION ≈ operation of interfacing entities that relies on observations not requiring the use of instrumentation, special test equipment, or subsequent analysis.

TEST ≈ operation of interfacing entities using instrumentation or special test equipment to collect data for later analysis.

ANALYSIS ≈ processing of accumulated or generated data obtained from other qualification methods; examples are reduction, calculation, simulation, interpretation, or extrapolation.

INSPECTION ≈ visual examination of interfacing entities, documentation, etc.

SPECIAL QUALIFICATION METHODS ≈ any special qualification means applied to the interfacing entities, such as special tools, techniques, procedures, facilities, and acceptance limits.

# 6. APPENDIXES

## 6.1. Data Structures and Element Details

## 6.2. Interface Mechanism Examples

- ELECTRICAL/FUNCTIONAL INTERFACE
- MECHANICAL/PHYSICAL INTERFACE
- SOFTWARE INTERFACE
- COMPATIBILITY ANALYSIS

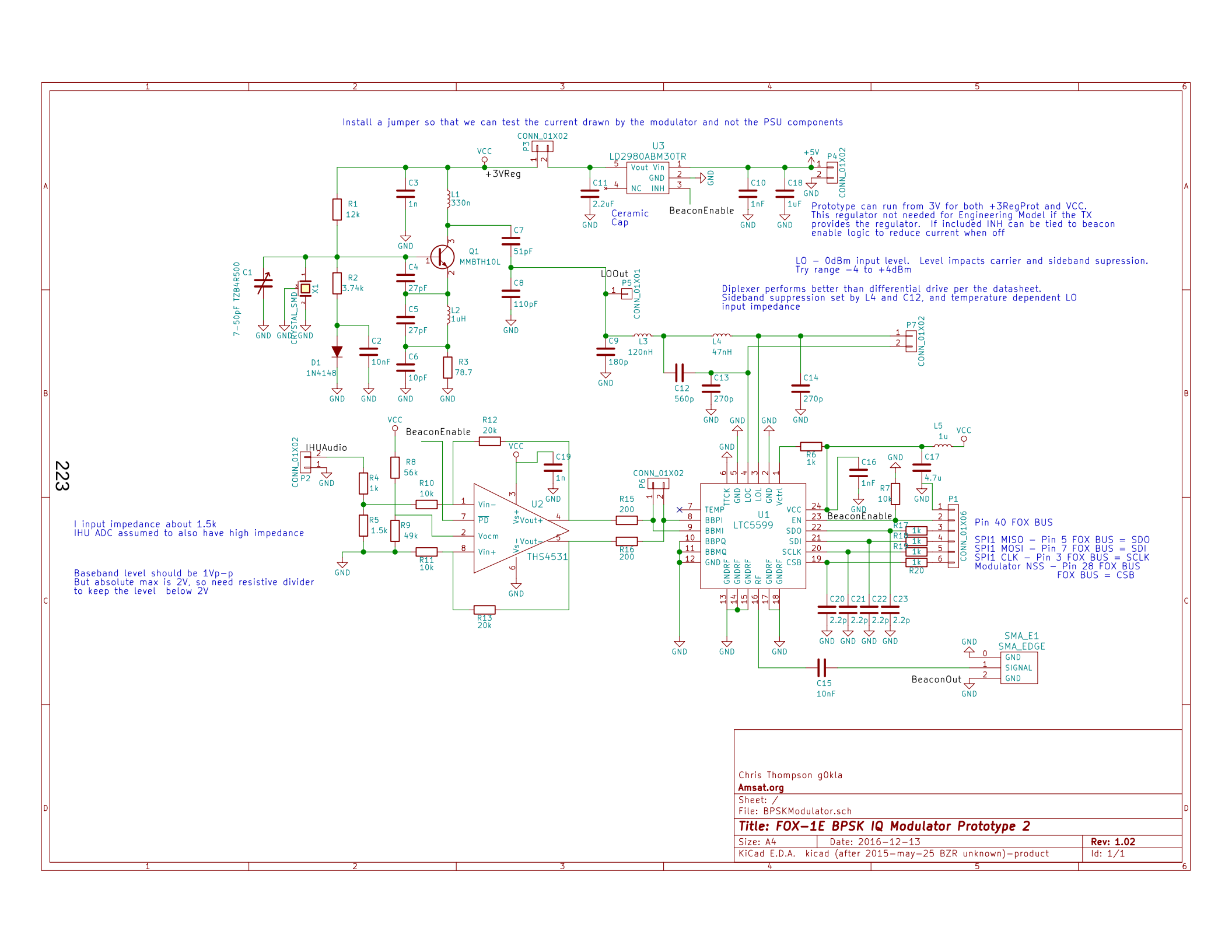
ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

## 6.3. Acronyms and Abbreviations

## 6.4. References/Applicable Documents

1) AMSAT Fox-1 Concept of Operations, Version 1.03, October 19, 2011
2) AMSAT RadFXSat (Fox-1E) System Requirements Specification (SRS), v1.01, 4 April 2016
3) CubeSat Design Specification, Rev. 13, February 20, 2014, The CubeSat Program Cal Poly SLO
4) ITU Radio Regulations, Edition of 2012, available from http://www.itu.int/publ/R-REG-RR-2012/en
5) Fox-1E Linear Transponder Communications Performance Analysis (Preliminary), AMSAT, DRAFT v0.3, 7 August 2016

ITAR CONTROLLED INFORMATION – FOR OFFICIAL USE ONLY
VANDERBILT UNIVERSITY and AMSAT PROPRIETARY
DISTRIBUTION WITHOUT VANDERBILT and AMSAT CONSENT IS PROHIBITED.

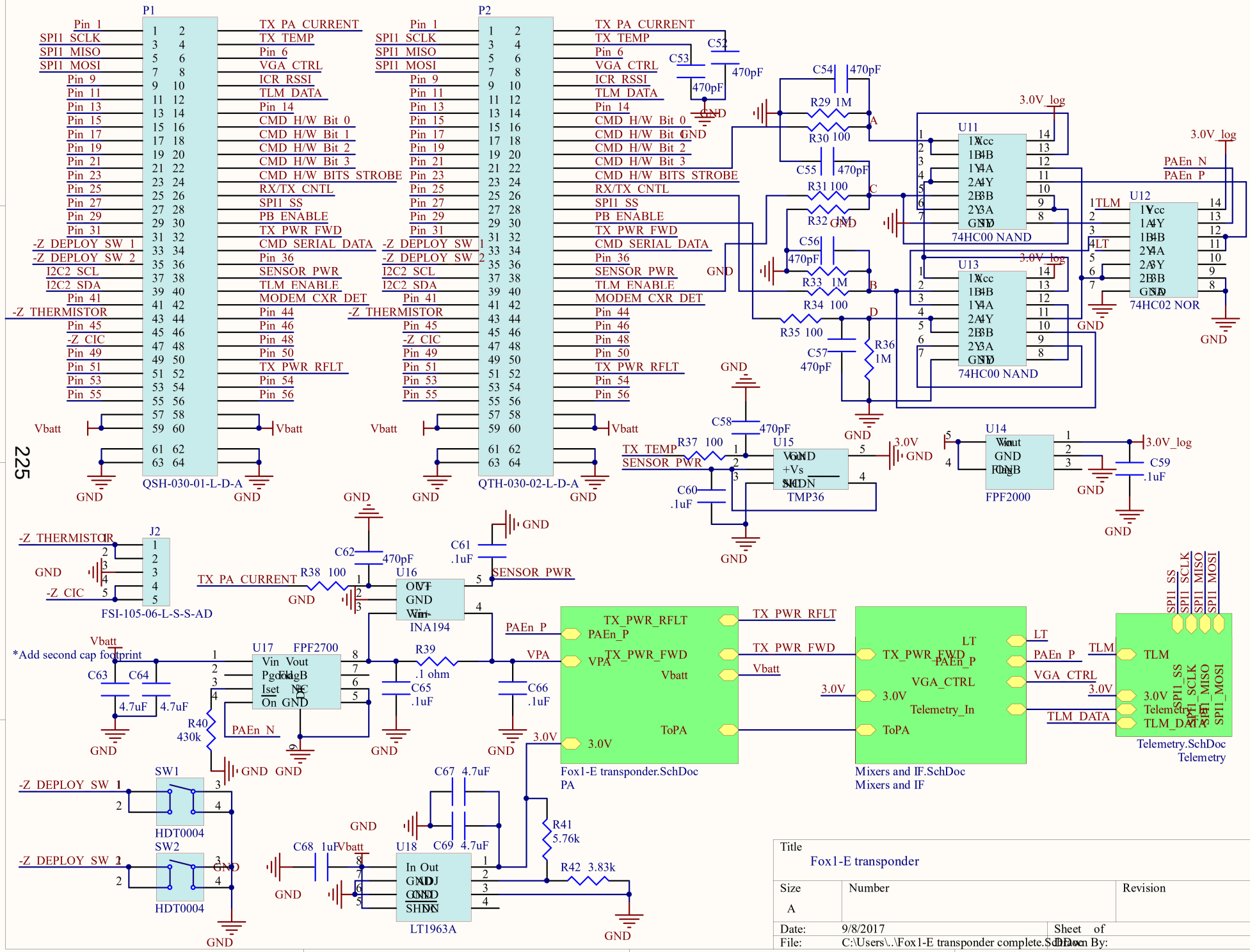


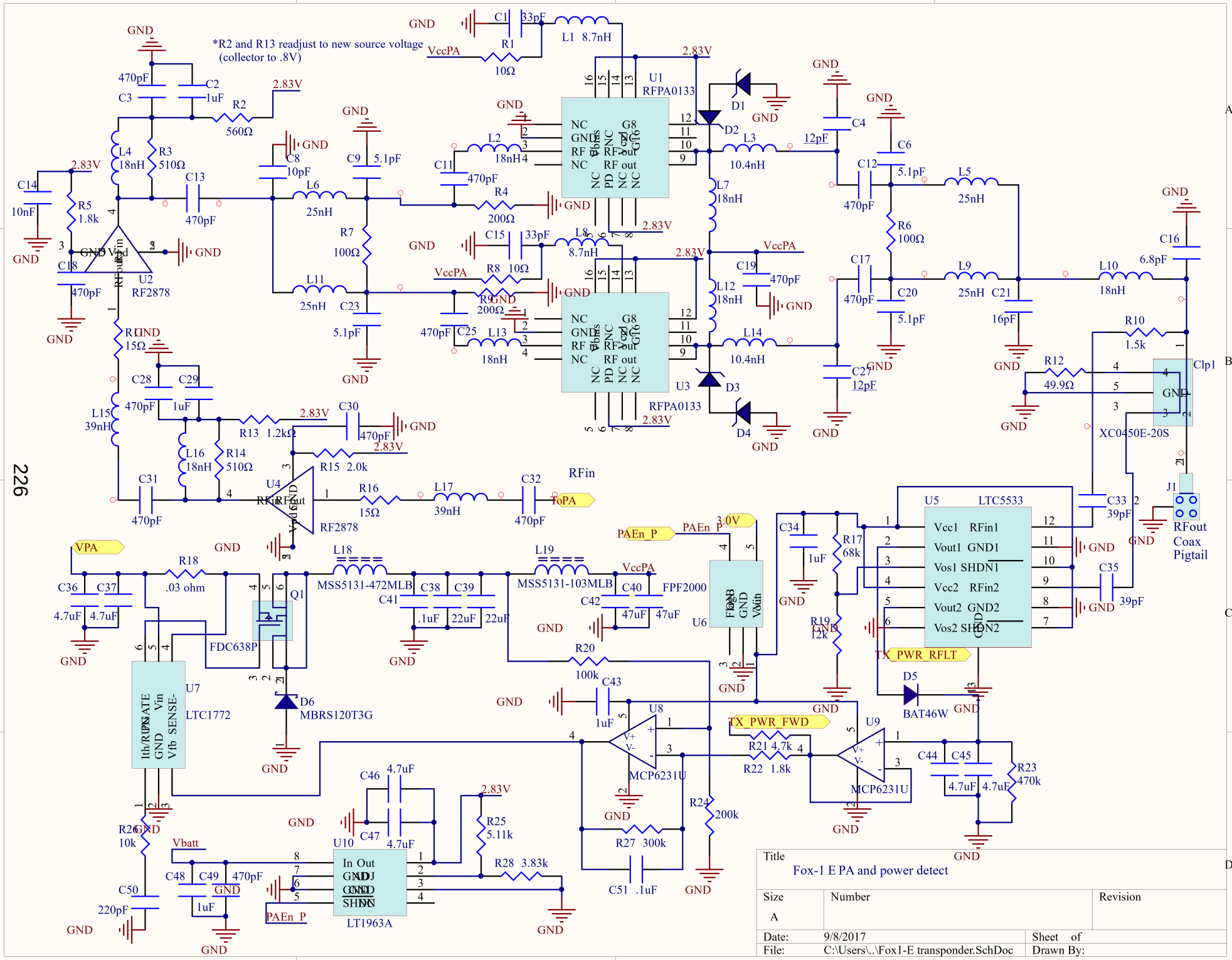

223

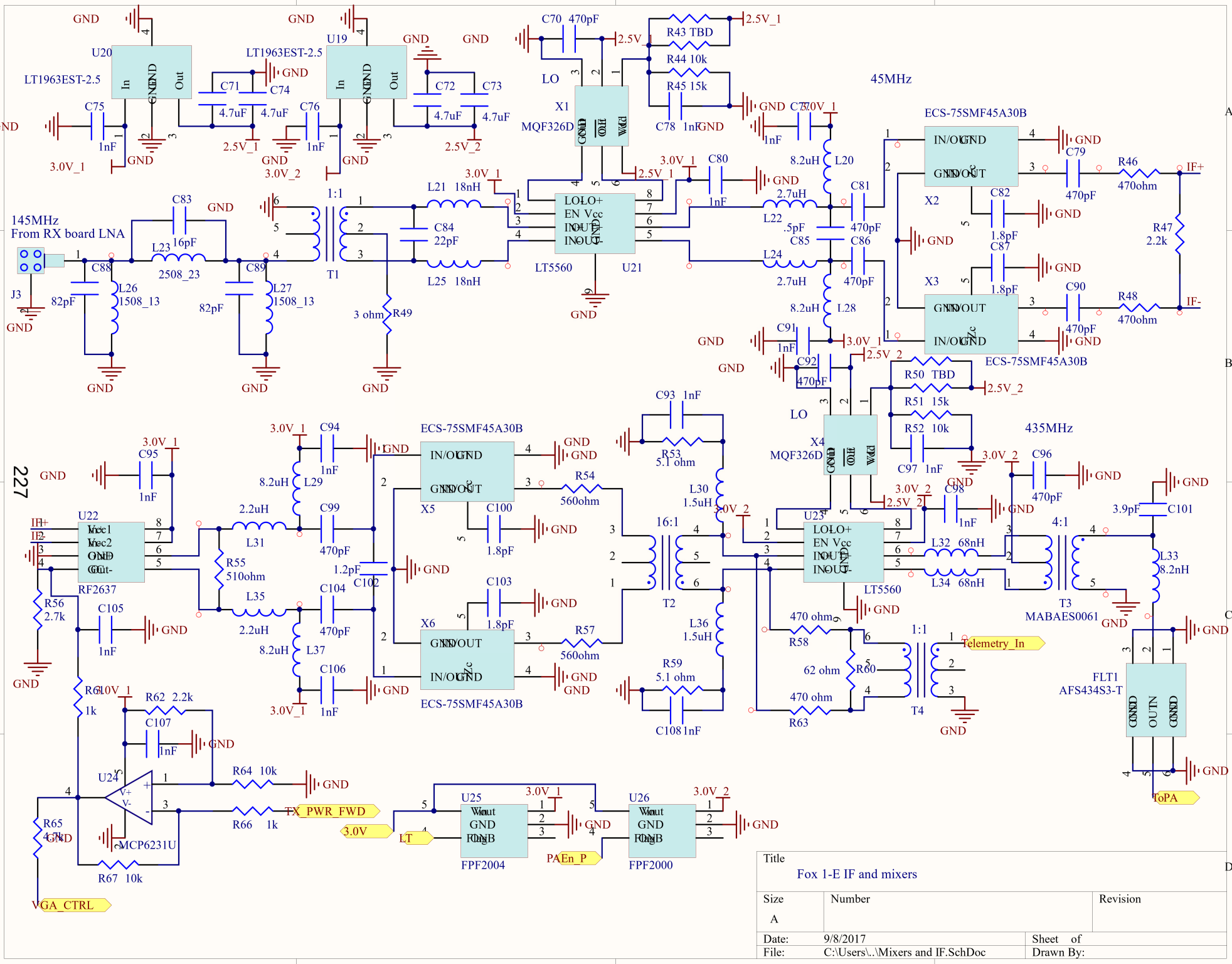
Chris Thompson g0kla
**Amsat.org**
Sheet: /
File: BPSKModulator.sch

***Title: FOX–1E BPSK IQ Modulator Prototype 2***

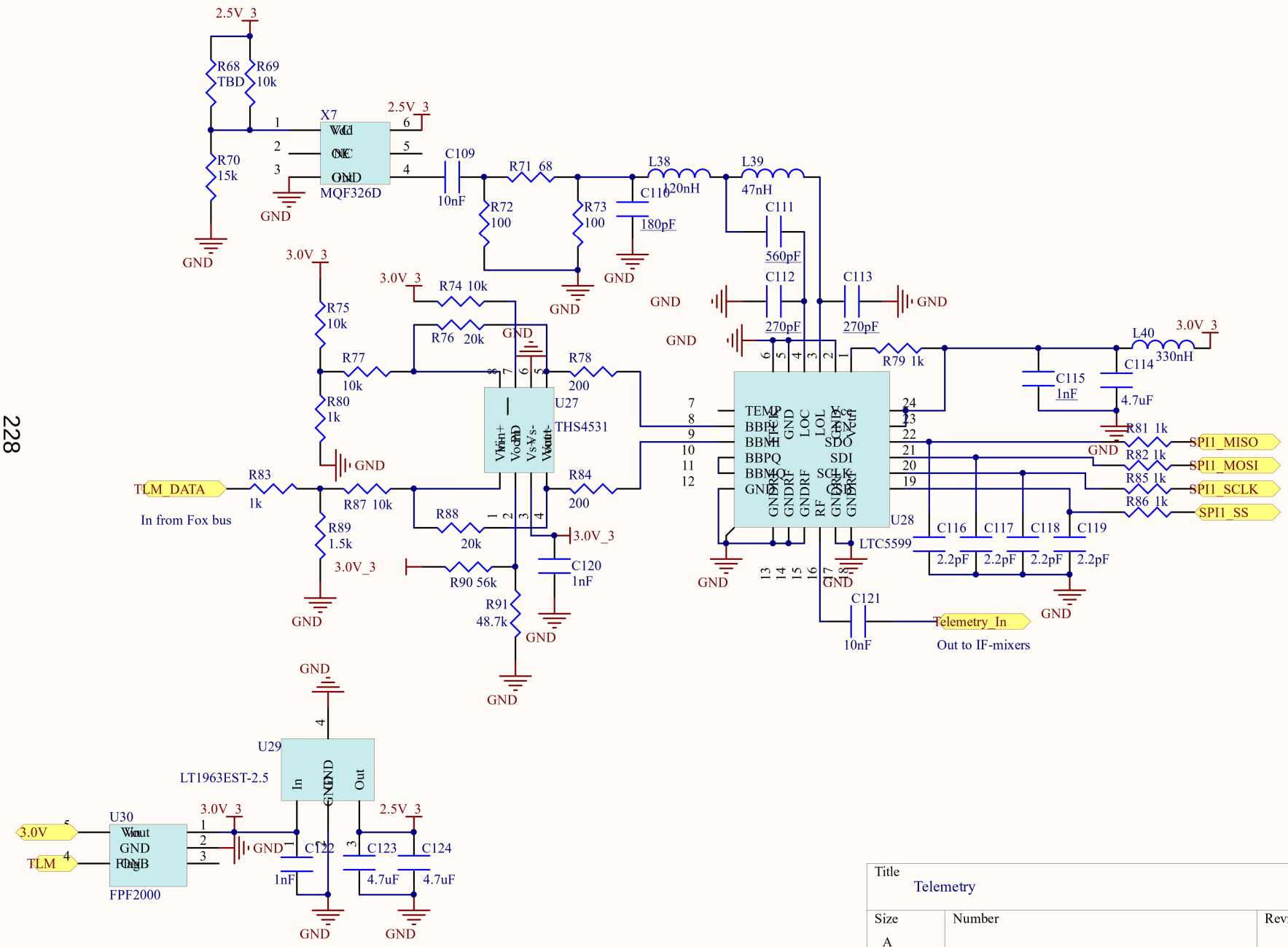
| Size: A4 | Date: 2016–12–13 | **Rev: 1.02** |
|---|---|---|
| KiCad E.D.A. kicad (after 2015–may–25 BZR unknown)–product | | Id: 1/1 |


P1
QSH-030-01-L-D-A
P2
QTH-030-02-L-D-A
TX_PA_CURRENT
TX_TEMP
VGA_CTRL
ICR_RSSI
TLM_DATA
CMD_H/W_Bit_0
CMD_H/W_Bit_1
CMD_H/W_Bit_2
CMD_H/W_Bit_3
CMD_H/W_BITS_STROBE
RX/TX_CNTL
SPI1_SS
PB_ENABLE
TX_PWR_FWD
CMD_SERIAL_DATA
SENSOR_PWR
TLM_ENABLE
MODEM_CXR_DET
TX_PWR_RFLT
SPI1_SCLK
SPI1_MISO
SPI1_MOSI
I2C2_SCL
I2C2_SDA
-Z_THERMISTOR
-Z_CIC
-Z_DEPLOY_SW_1
-Z_DEPLOY_SW_2
Vbatt
GND
3.0V_log
PAEn_N
PAEn_P
74HC00 NAND
74HC02 NOR
TMP36
FPF2000
FSI-105-06-L-S-S-AD
INA194
FPF2700
*Add second cap footprint
HDT0004
LT1963A
Fox1-E transponder.SchDoc
PA
Mixers and IF.SchDoc
Mixers and IF
Telemetry.SchDoc
Telemetry
Title
Fox1-E transponder
Size
A
Number
Revision
Date: 9/8/2017
Sheet of
File: C:\Users\..\Fox1-E transponder complete.SchDoc
Drawn By:


*R2 and R13 readjust to new source voltage
(collector to .8V)
U1 RFPA0133
U2 RF2878
U3 RFPA0133
U4 RF2878
U5 LTC5533
U6 FPF2000
U7 LTC1772
U8 MCP6231U
U9 MCP6231U
U10 LT1963A
Q1 FDC638P
D6 MBRS120T3G
D5 BAT46W
Clp1 XC0450E-20S
J1 RFout Coax Pigtail
RFin
ToPA
VPA
PAEn_P
TX_PWR_RFLT
TX_PWR_FWD
Vbatt
VccPA
2.83V
3.0V
GND
Title
Fox-1 E PA and power detect
Size
A
Number
Revision
Date: 9/8/2017
Sheet of
File: C:\Users\..\Fox1-E transponder.SchDoc
Drawn By:

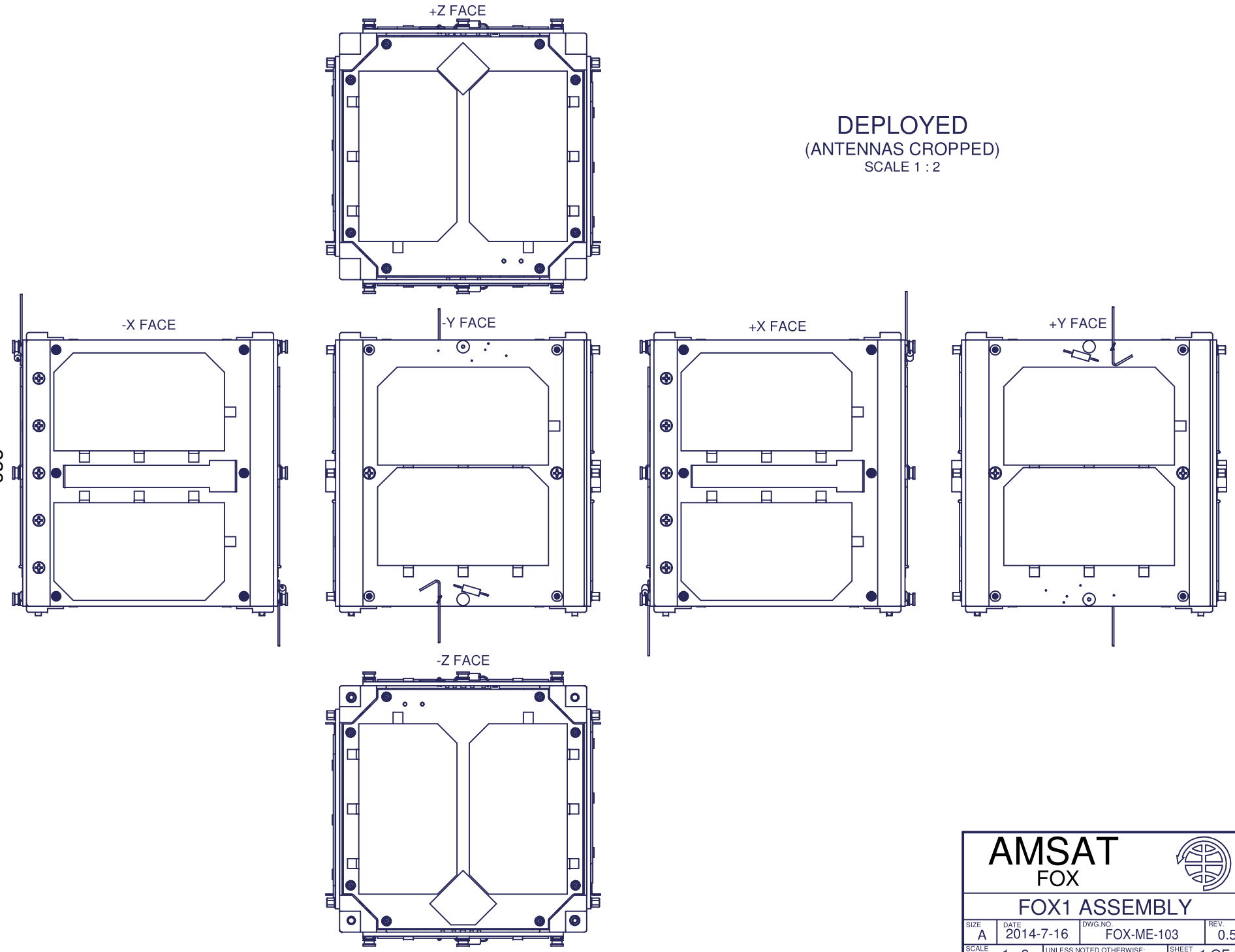


145MHz
From RX board LNA
J3
U20
LT1963EST-2.5
U19
LT1963EST-2.5
C75
1nF
C71
4.7uF
C74
4.7uF
C76
1nF
C72
4.7uF
C73
4.7uF
C83
16pF
C88
82pF
L23
2508_23
L26
1508_13
C89
82pF
L27
1508_13
1:1
T1
3 ohm
R49
C84
22pF
L21 18nH
L25 18nH
C70 470pF
LO
X1
MQF326D
LT5560
U21
R43 TBD
R44 10k
R45 15k
C78 1nF
C80
1nF
C77
1nF
8.2uH
L20
2.7uH
L22
.5pF
C85
C81
470pF
C86
470pF
L24
2.7uH
8.2uH
L28
C91
1nF
C92
470pF
45MHz
ECS-75SMF45A30B
X2
X3
C79
470pF
C82
1.8pF
C87
1.8pF
C90
470pF
R46
470ohm
R47
2.2k
R48
470ohm
IF+
IF-
R50 TBD
R51 15k
R52 10k
C97 1nF
435MHz
X4
MQF326D
C96
470pF
C98
1nF
U23
LT5560
L32 68nH
L34 68nH
4:1
T3
MABAES0061
3.9pF
C101
L33
8.2nH
FLT1
AFS434S3-T
ToPA
C93 1nF
R53
5.1 ohm
L30
1.5uH
16:1
T2
L36
1.5uH
R59
5.1 ohm
C108 1nF
470 ohm
R58
62 ohm
R60
470 ohm
R63
1:1
T4
Telemetry_In
C95
1nF
U22
RF2637
R56
2.7k
C105
1nF
R61
1k
R62 2.2k
C107
1nF
U24
MCP6231U
R64 10k
R66 1k
R65
4.7k
R67 10k
VGA_CTRL
TX_PWR_FWD
3.0V
LT
U25
FPF2004
U26
FPF2000
PAEn_P
C94
1nF
8.2uH
L29
2.2uH
L31
C99
470pF
R55
510ohm
L35
2.2uH
C104
470pF
8.2uH
L37
C106
1nF
1.2pF
C102
C100
1.8pF
C103
1.8pF
R54
560ohm
R57
560ohm
ECS-75SMF45A30B
X5
X6
ECS-75SMF45A30B
227
Title
Fox 1-E IF and mixers
Size
A
Number
Revision
Date: 9/8/2017
Sheet of
File: C:\Users\..\Mixers and IF.SchDoc
Drawn By:




228

| | | |
|---|---|---|
| Title: Telemetry | | |
| Size: A | Number: | Revision: |
| Date: 9/8/2017 | | Sheet of |
| File: C:\Users\..\Telemetry.SchDoc | | Drawn By: |




DEPLOYED
(ANTENNAS CROPPED)
SCALE 1 : 2

229

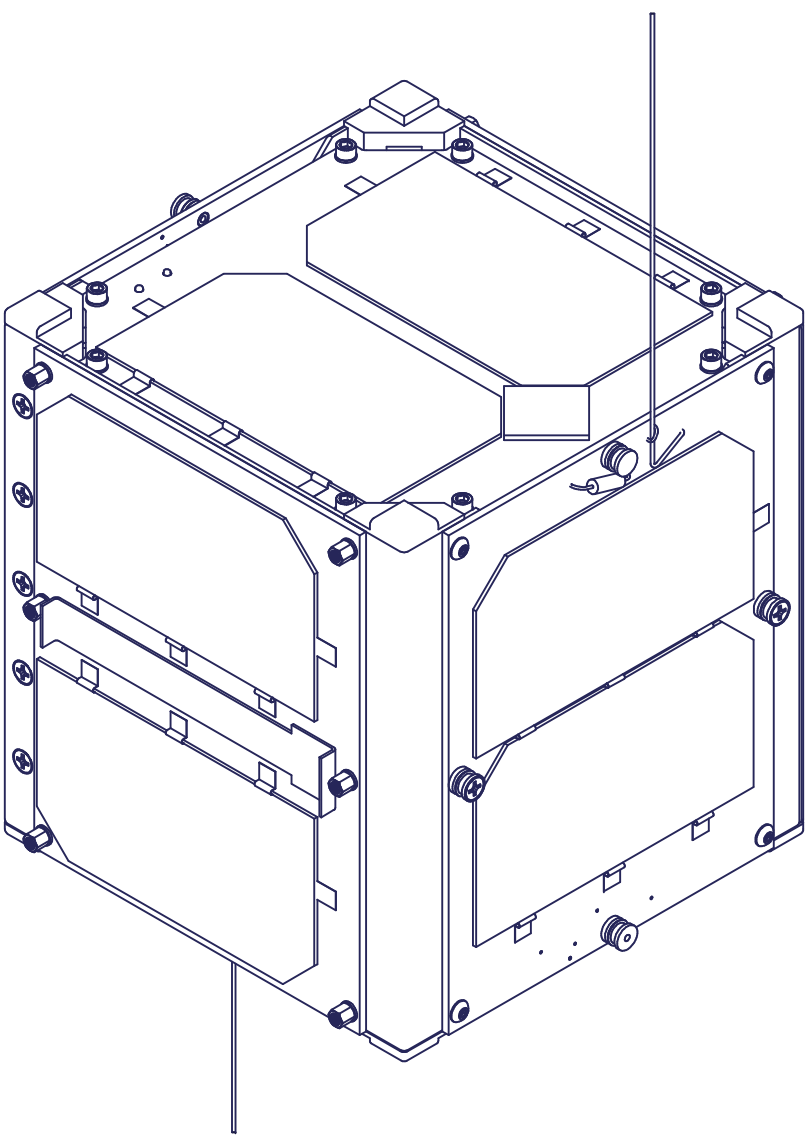
AMSAT
FOX

FOX1 ASSEMBLY

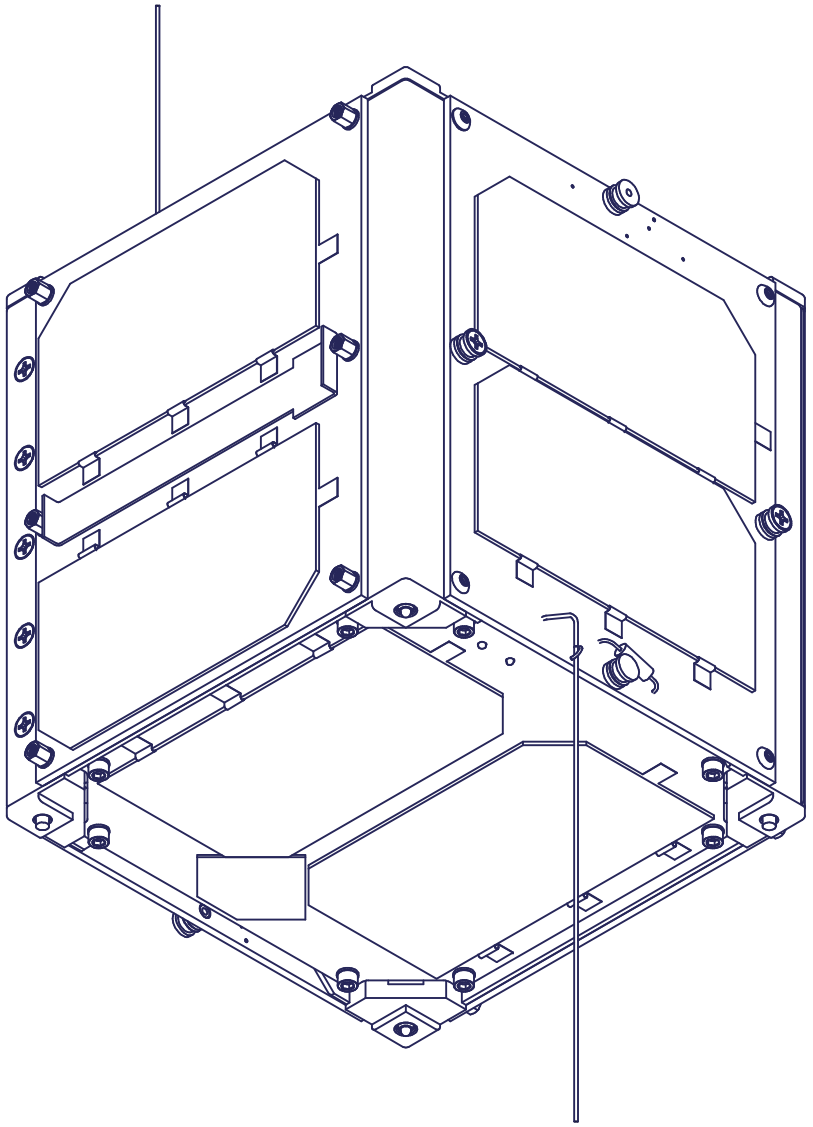
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-7-16 | FOX-ME-103 | 0.5 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 5 |

ISO +X+Y+Z
SCALE 3 : 4



ISO -X-Y-Z
SCALE 3 : 4



230

# AMSAT
FOX

## FOX1 ASSEMBLY

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-7-16 | FOX-ME-103 | 0.5 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 3 : 4 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 2 OF 5 |

+X FACE
TRUE ANTENNA LENGTHS
SCALE 0.2 : 1

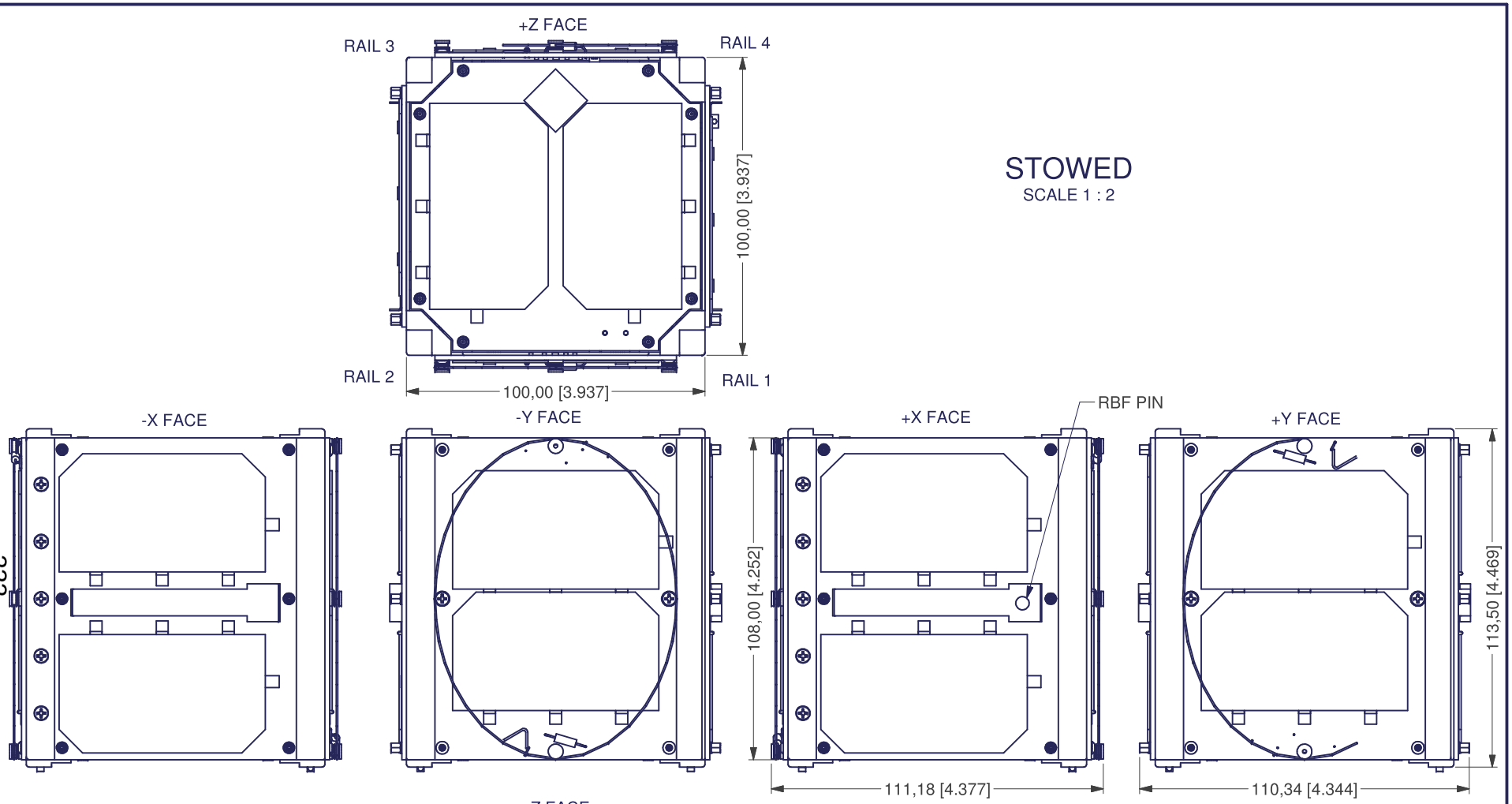
231

AMSAT
FOX

FOX1 ASSEMBLY

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-7-16 | FOX-ME-103 | 0.5 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 5 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 3 OF 5 |



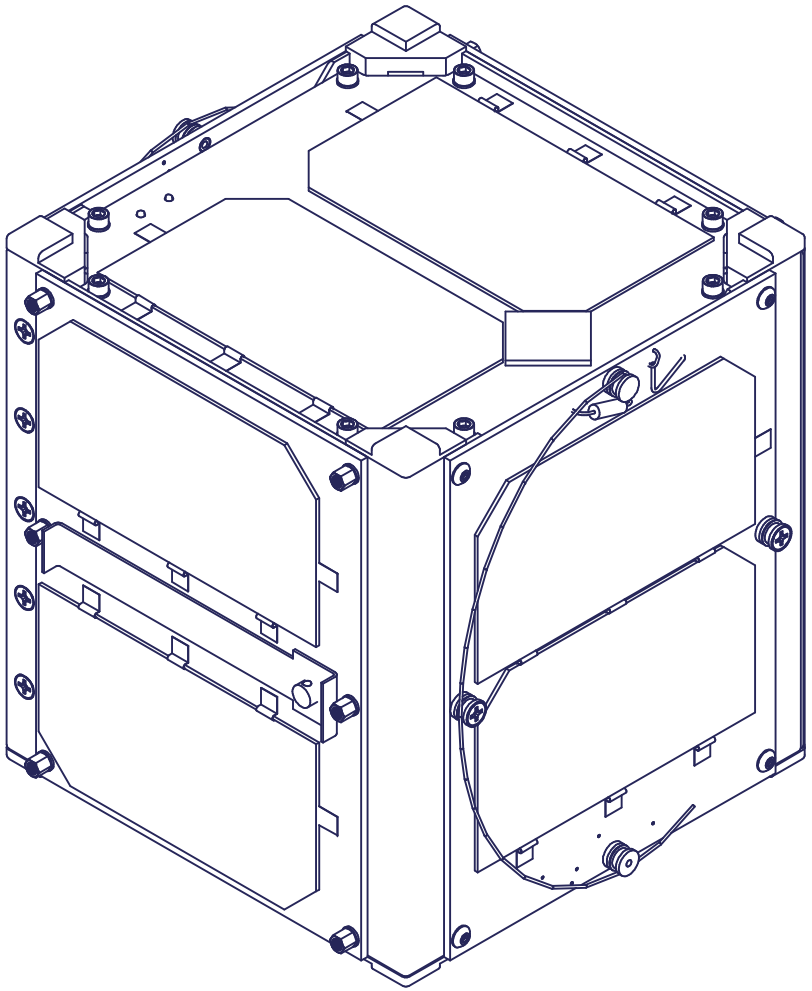

STOWED

SCALE 1 : 2

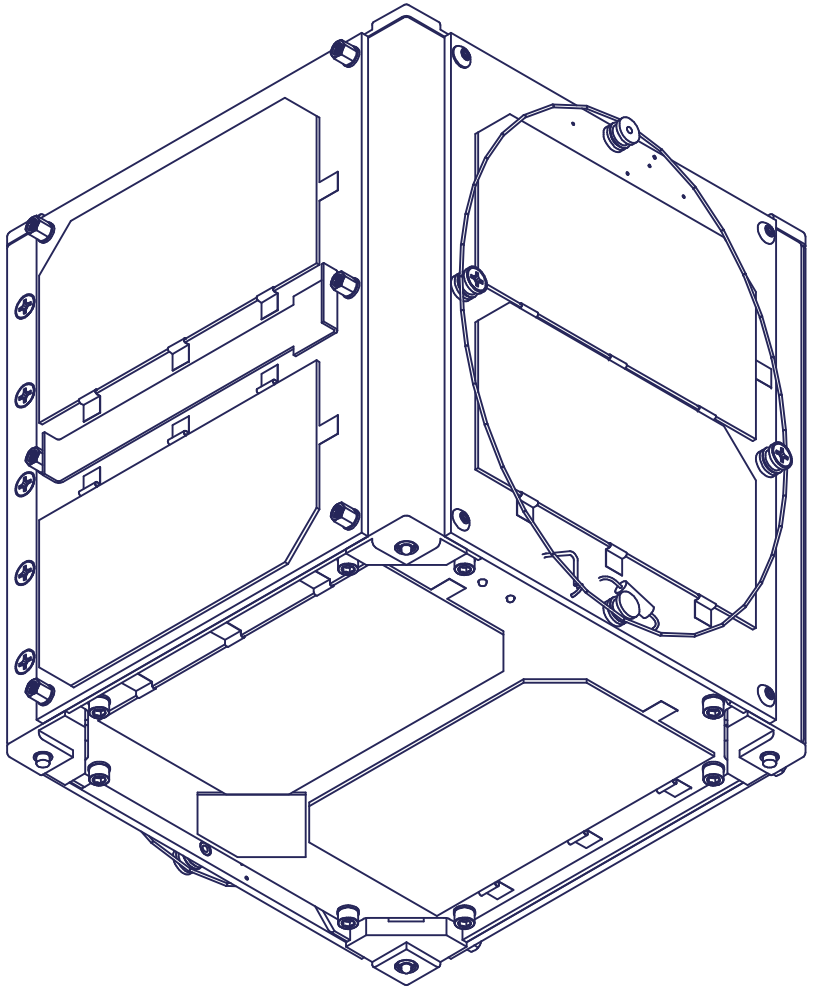
232

| | | | |
|---|---|---|---|
| AMSAT FOX | | | |
| FOX1 ASSEMBLY | | | |
| SIZE A | DATE 2014-7-16 | DWG.NO. FOX-ME-103 | REV. 0.5 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | | SHEET 4 OF 5 |

ISO +X+Y+Z
SCALE 3 : 4



ISO -X-Y-Z
SCALE 3 : 4

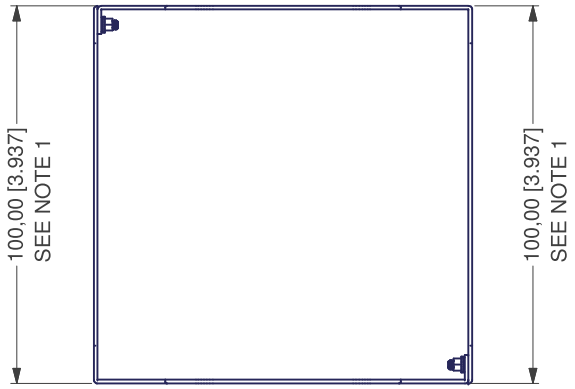


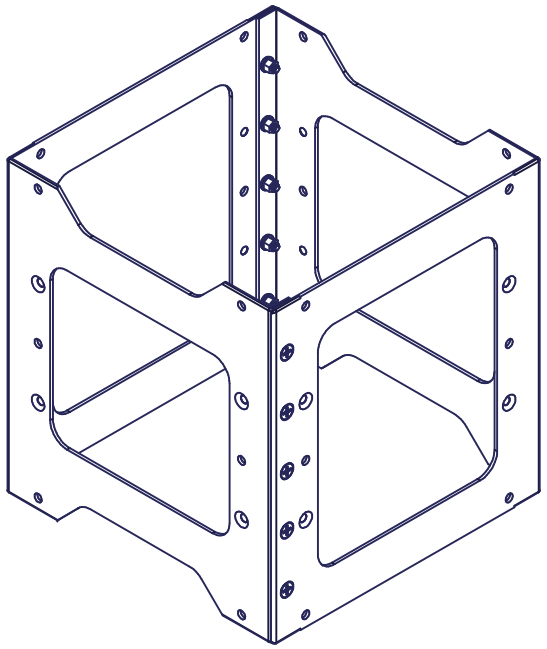


AMSAT
FOX

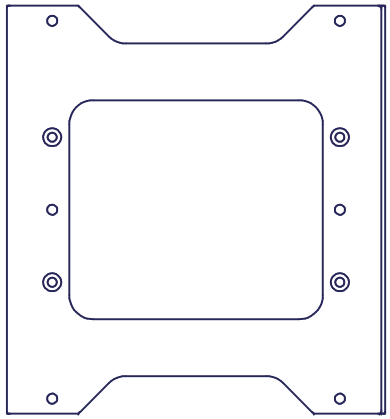
FOX1 ASSEMBLY

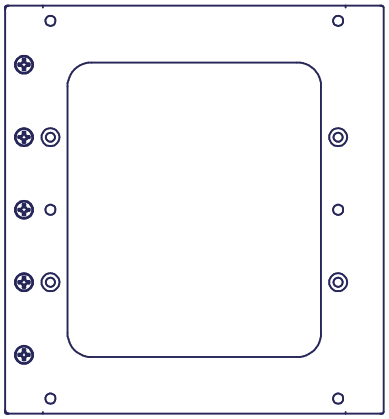
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-7-16 | FOX-ME-103 | 0.5 |
| SCALE 3 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | | SHEET 5 OF 5 |



-01 WALLS





| ITEM | QTY | PART NUMBER | DESCRIPTION |
|---|---|---|---|
| 1 | 2 | FOX-ME-111-03 | WALLS, BENT |
| 3 | 10 | MS21042-02 | LOCK NUT, #2-56 |
| 4 | 10 | NAS662C2R4 / MCMASTER-CARR 93085A017 | FHCS, #2-56 X 1/4" X 100° SS |
| 5 | 10 | 90945A705 | WASHER, #2 THIN, NAS620C2 |

Parts List

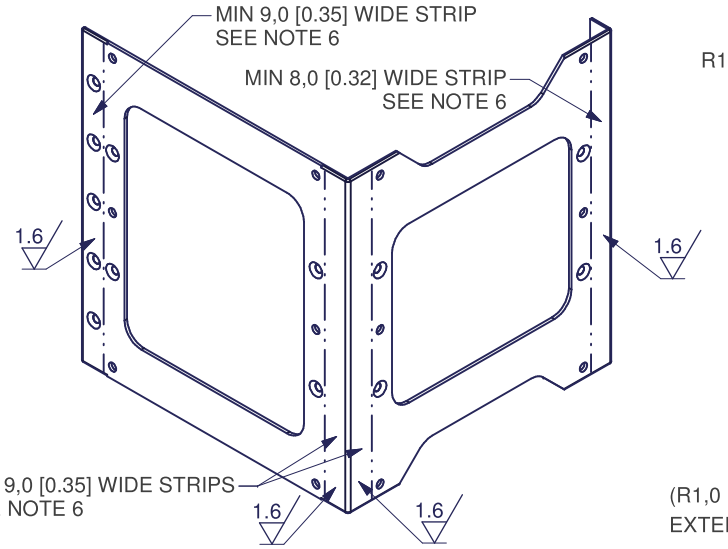
NOTES:
1) AT ASSEMBLY, ALIGN BOTH DIMENSIONS. TORQUE #2-56 NUTS TO 2.5 IN-LB, OR 40 IN-OZ.
2. CONFORMAL COAT OR EPOXY COAT CAD-PLATED STEEL NUTS.

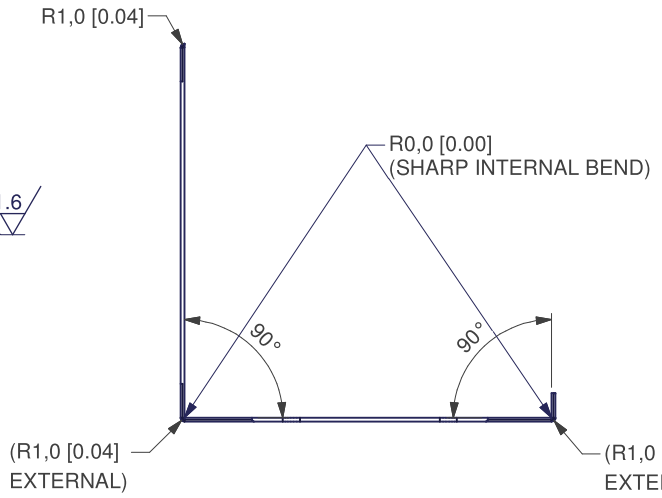
AMSAT
FOX

WALLS

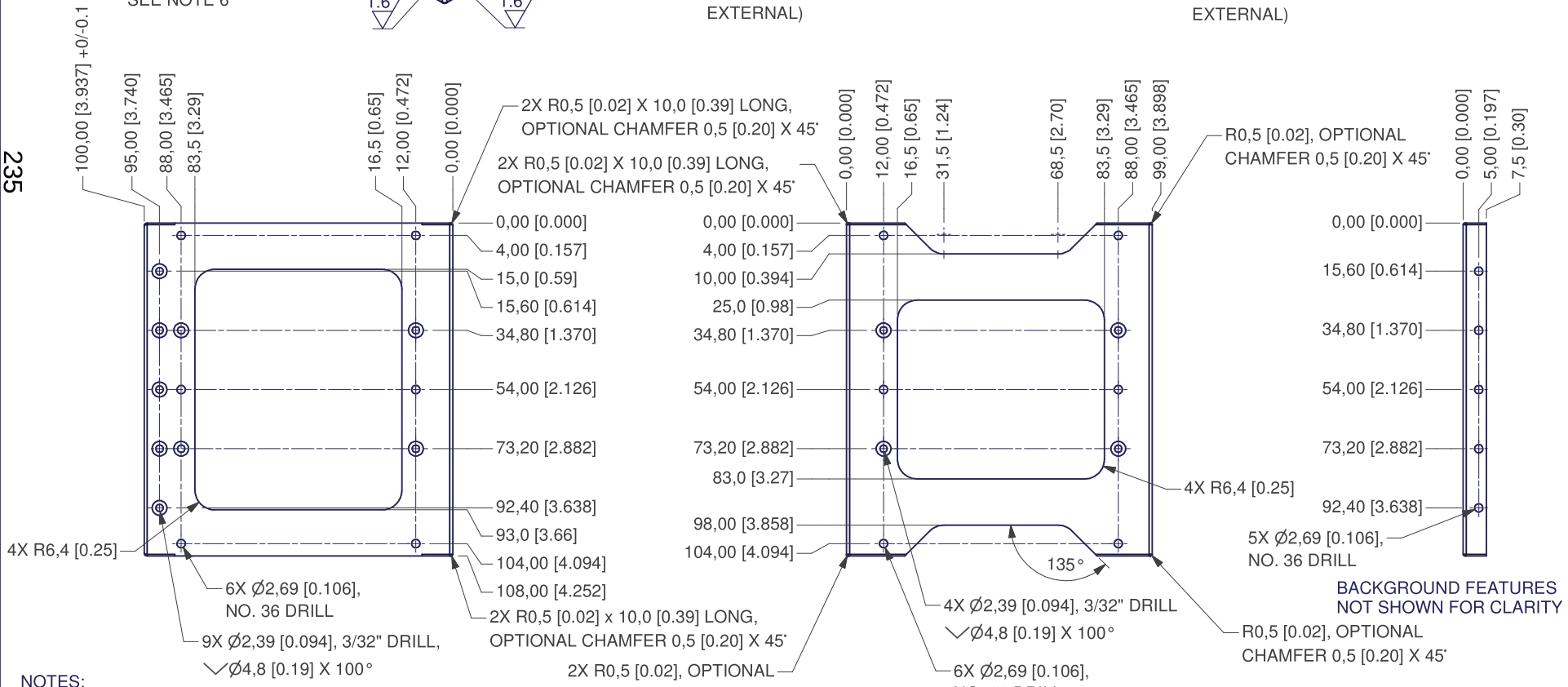
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-07-28 | FOX-ME-111 | 1 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 3 |



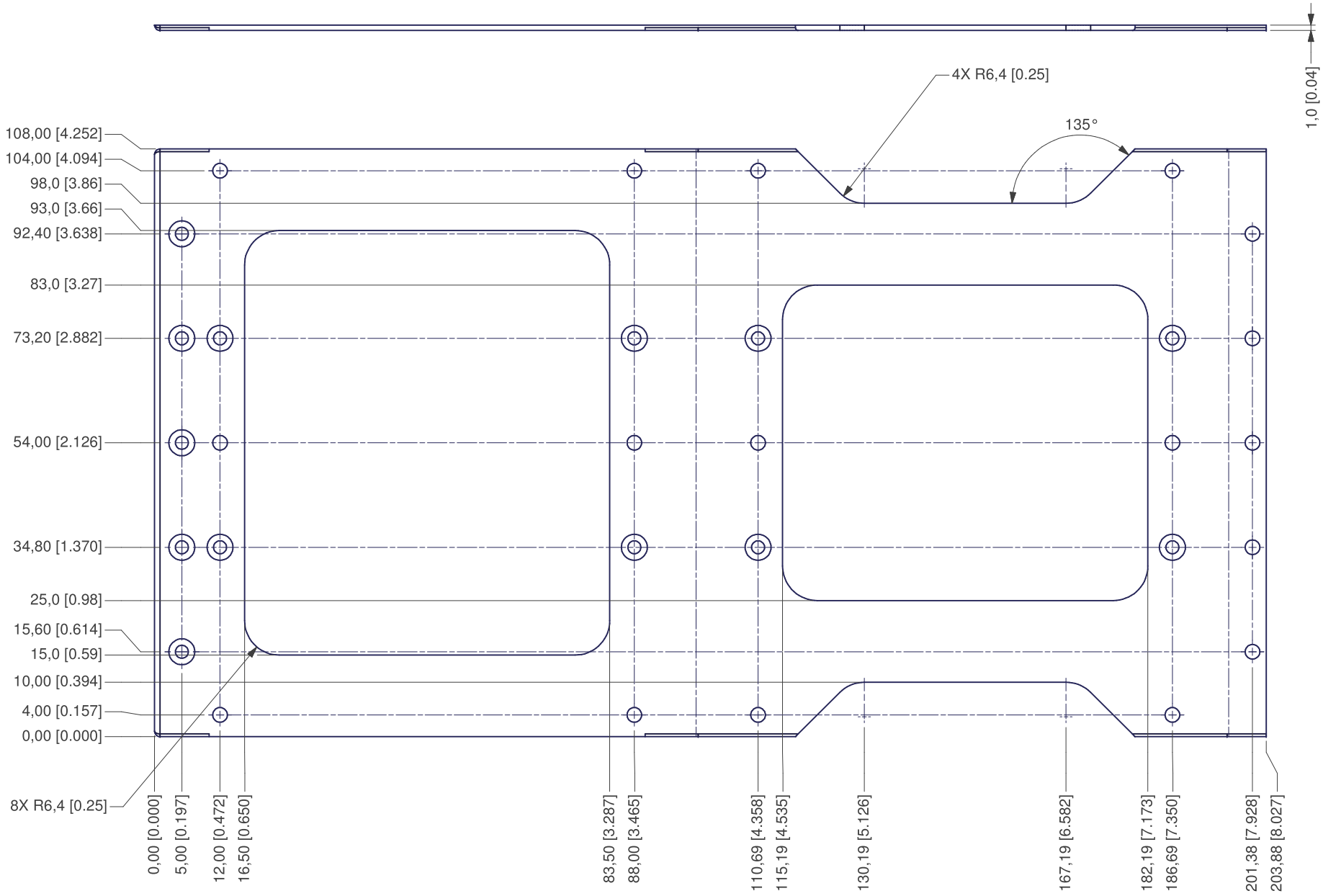




# -03 WALLS, BENT




NOTES:
3) MATERIAL: ALUMINUM 5052-H32.
4) DO NOT SCUFF.
5) CHEMICAL CONVERSION COAT PER MIL-DTL-5541 CLASS 3, ELECTRICALLY CONDUCTIVE CLEAR.
6) HARD ANODIZE AREAS SHOWN PER MIL-A-8625 TYPE III, CLASS 1, CLEAR.
INCLUDE BEND/ROUND, CONSUMED HOLES AND SHEETMETAL EDGES.

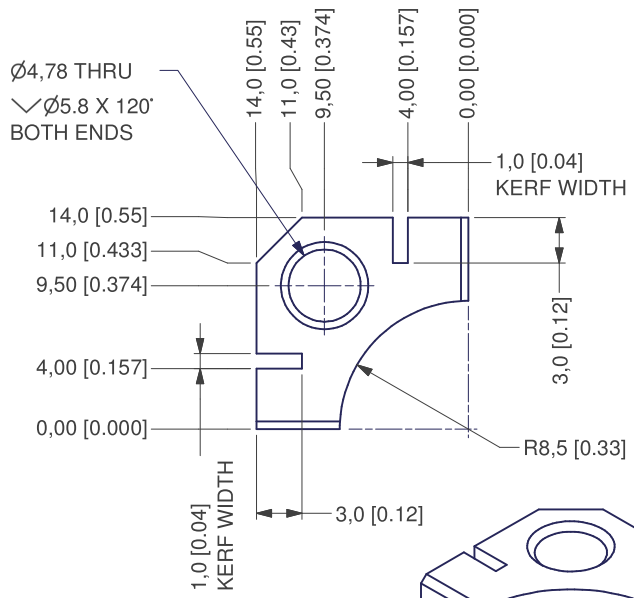
| WALLS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-28 | DWG.NO. FOX-ME-111 | REV. 1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | SHEET 2 OF 3 | |

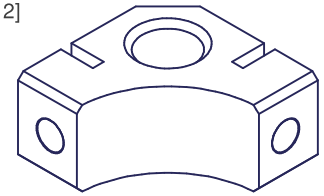


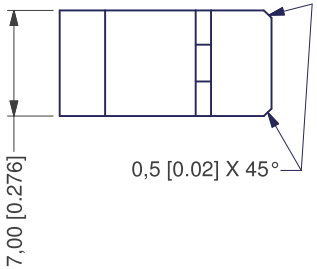

# -03 WALLS, FLAT PATTERN

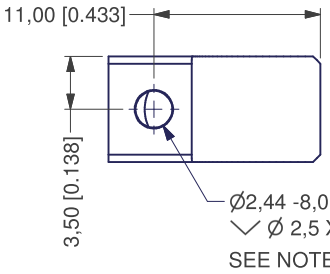
REFERENCE DIMENSIONS
POSITION/TOLERANCE FROM BENDS

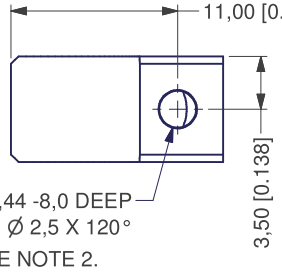
| WALLS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-28 | DWG.NO. FOX-ME-111 | REV. 1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 3 |

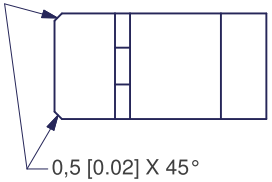


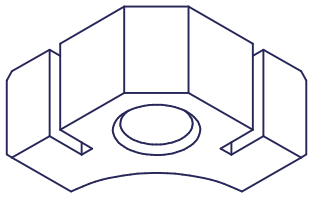





-01

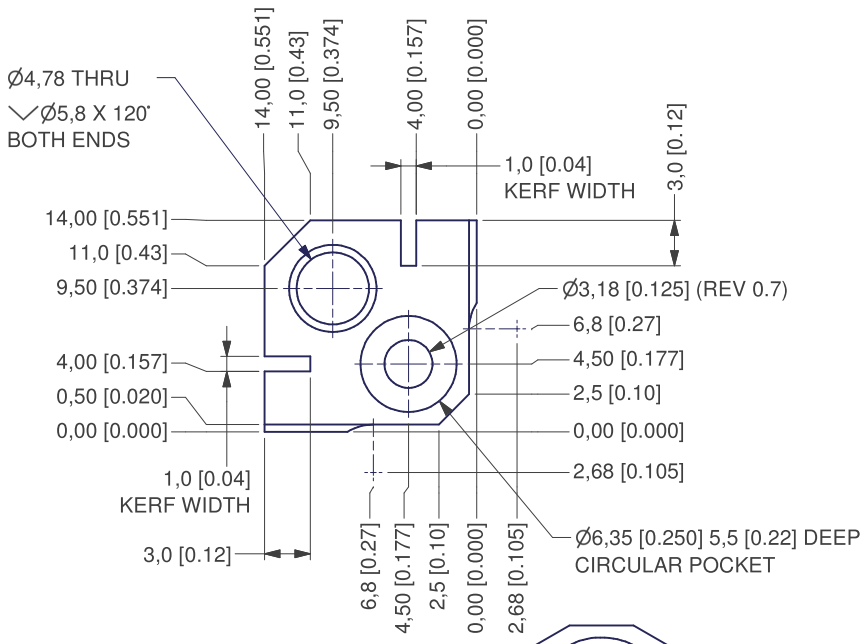
NOTES:
1) MATERIAL: BLACK DELRIN

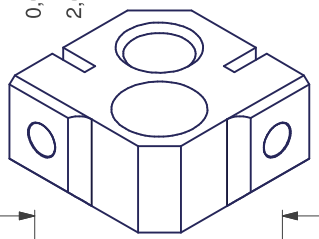
2) STI TAP #2-56 MIN ↧4,0 [0.16] , USING HELICOIL BOTTOMING TAP 02CBB.
INSTALL HELICOIL #2-56 X 1 DIA 3585-02CN0086,
USING HELICOIL INSTALLATION TOOL 551-02.
REMOVE TANG USING HELICOIL TANG-BREAKOFF TOOL 3695-02.

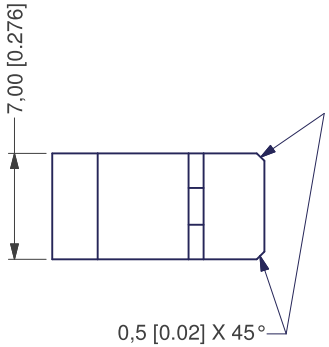
AMSAT
FOX

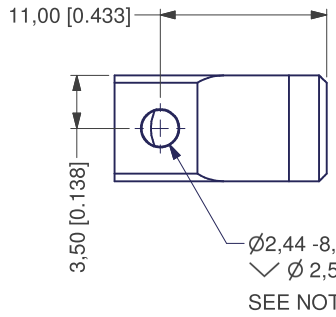
PCB DELRIN MOUNT

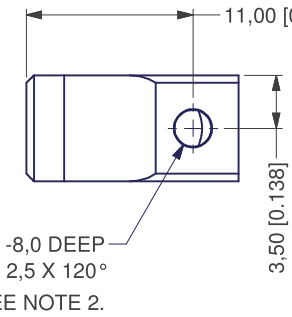
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-04-24 | FOX-ME-118 | 0.7 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | SHEET 1 OF 2 | |

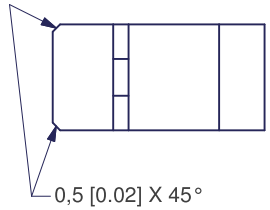


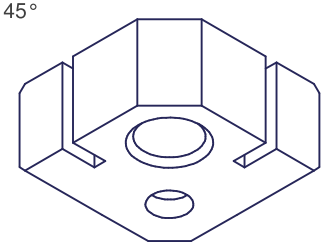





-03

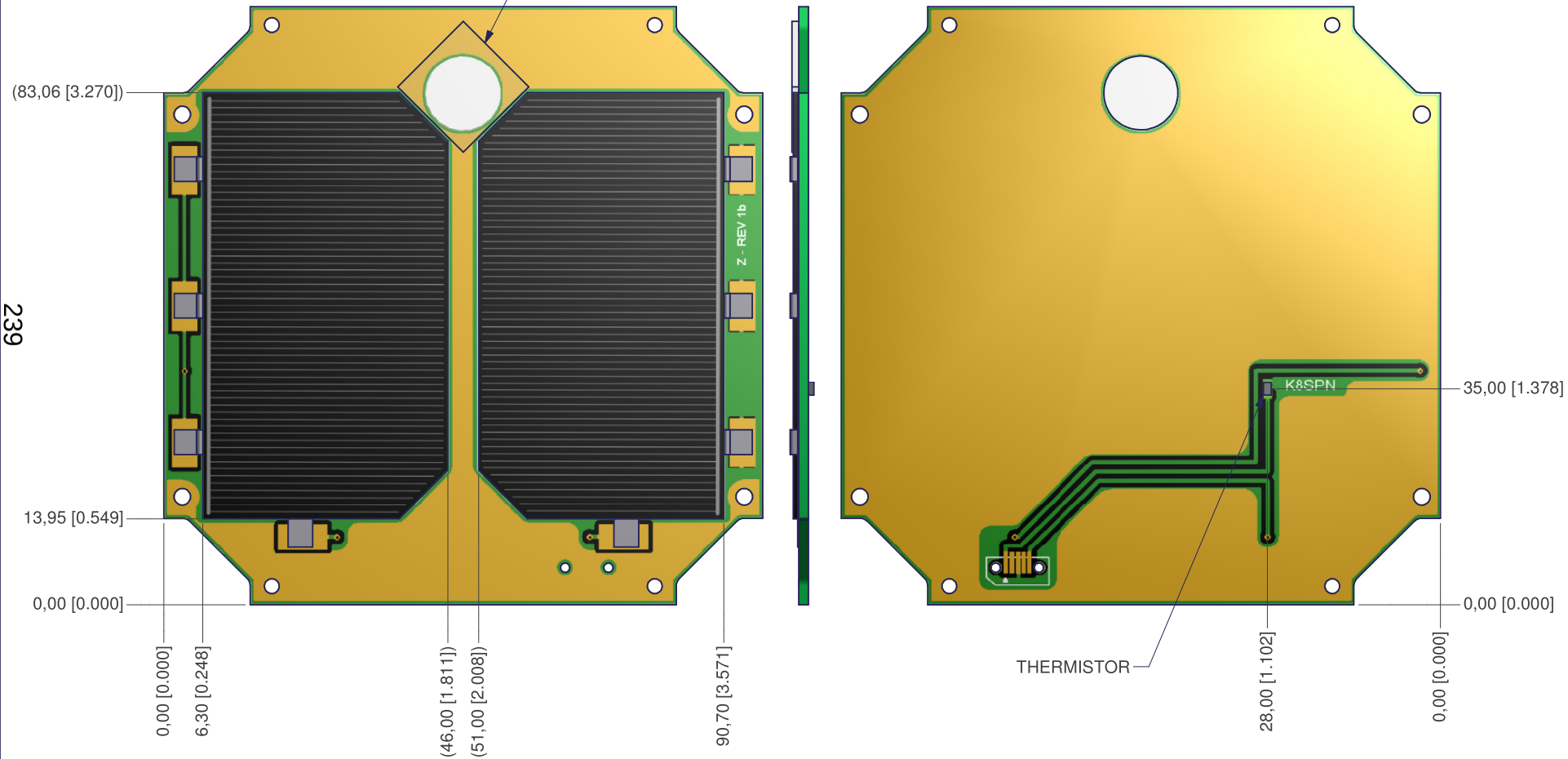
| PCB DELRIN MOUNT | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-24 | DWG.NO. FOX-ME-118 | REV. 0.7 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 2 |

# -01
# +Z TOP

TOP SIDE

BOTTOM SIDE

EDMUND OPTICS 53709 IR FILTER,
IR COATED SURFACE FACING SPACECRAFT EXTERIOR.
BOND USING LOCTITE 9340 EPOXI-PATCH.




| ITEM | QTY | PART NUMBER | DESCRIPTION |
|---|---|---|---|
| 1 | 1 | FOX-ME-120-03 | PCB |
| 6 | 2 | FOX-ME-122 | SpaceQuest UTJ |
| 7 | 1 | | THERMISTOR, 0805 |
| 9 | 1 | | IR FILTER |

Parts List

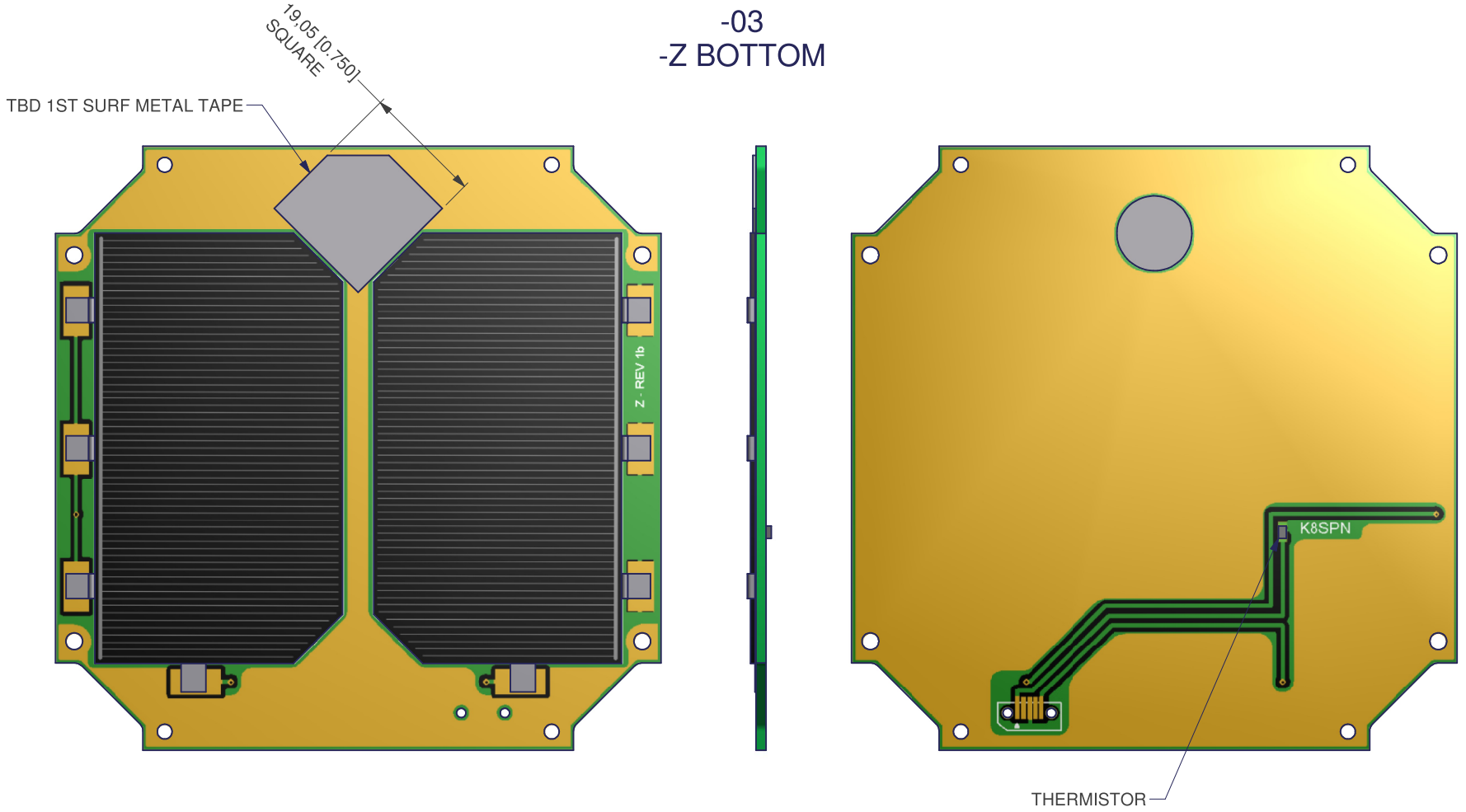
AMSAT
FOX

Z TOP/BOTT SOLAR PANEL

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-08-21 | FOX-ME-120 | 1.0 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 3 |

# -03
# -Z BOTTOM




TOP SIDE

BOTTOM SIDE

240

| ITEM | QTY | PART NUMBER | DESCRIPTION |
|---|---|---|---|
| 1 | 1 | FOX-ME-120-03 | PCB |
| 6 | 2 | FOX-ME-122 | SpaceQuest UTJ |
| 7 | 1 | | THERMISTOR, 0805 |
| 8 | 1 | | 1ST SURF METAL TAPE |

Parts List

AMSAT
FOX

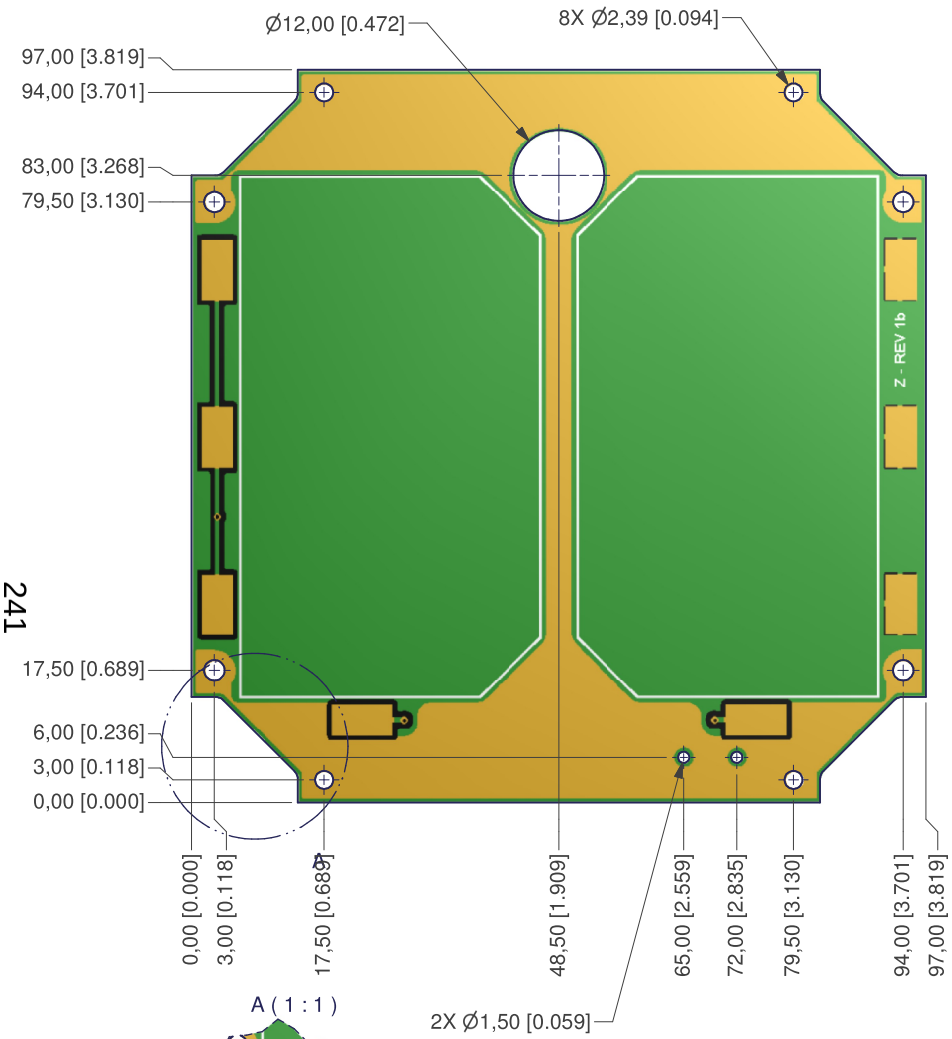
Z TOP/BOTT SOLAR PANEL

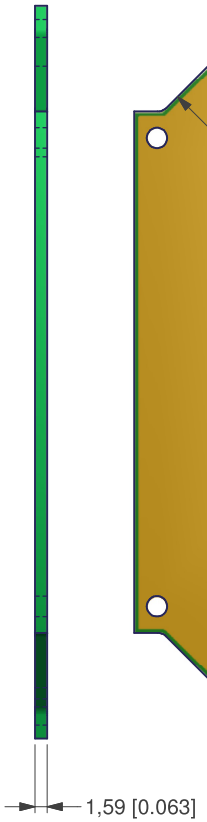
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-08-21 | FOX-ME-120 | 1.0 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 2 OF 3 |

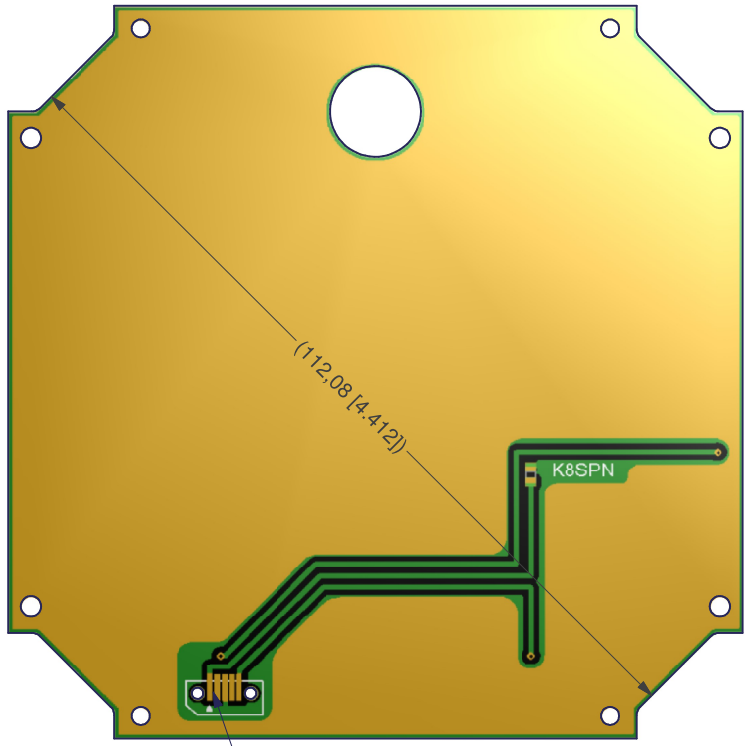
-05
PCB

TOP SIDE

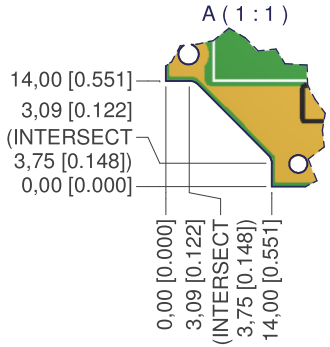



BOTTOM SIDE



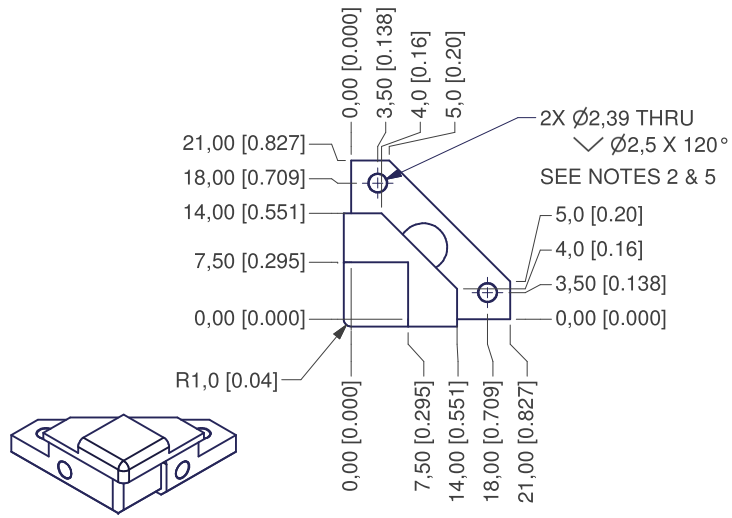

A ( 1 : 1 )

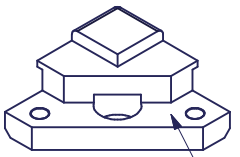


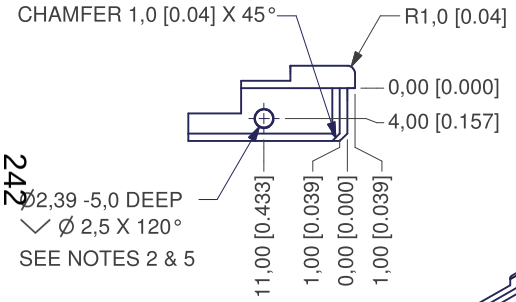

NOTES:
1) PLATING FINISH: FLASH GOLD TOP & BOTTOM SIDES.
2) BOTTOM SIDE, SELECTIVE GOLD FIVE PADS OF CONNECTOR 30 MILS.

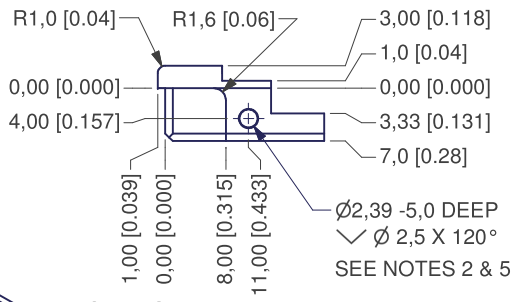
241

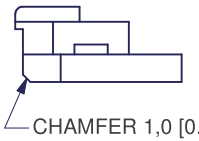
| | | | |
|---|---|---|---|
| AMSAT FOX | | | |
| Z TOP/BOTT SOLAR PANEL | | | |
| SIZE A | DATE 2014-08-21 | DWG.NO. FOX-ME-120 | REV. 1.0 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | | SHEET 3 OF 3 |

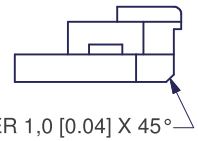


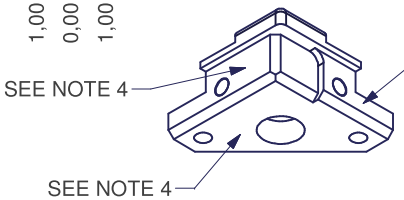



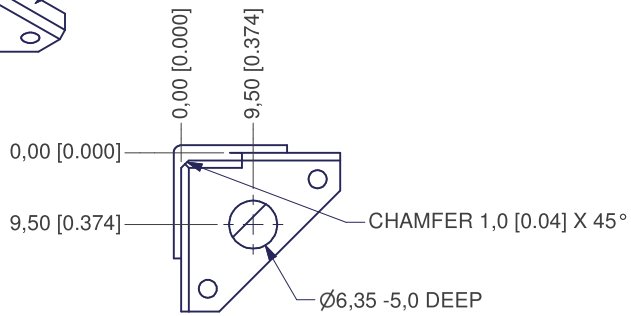




-01 +Z

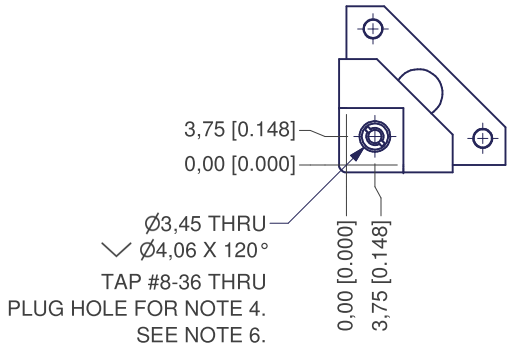


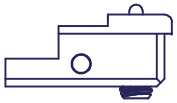

NOTES:
1) -01, -03, -09 MATERIAL: 6061-T6
2) STI TAP #2-56 MIN ↧4, USING HELICOIL BOTTOMING TAP 02CBB.
3) CHEMICAL CONVERSION COAT PER MIL-DTL-5541 CLASS 3, ELECTRICALLY CONDUCTIVE CLEAR.
4) MASK SURFACES AS SHOWN.
   HARD ANODIZE REMAINING SURFACES PER MIL-A-8625 TYPE III, CLASS 1, CLEAR.
5. INSTALL HELICOIL #2-56 X 1 DIA 358502CN0086,
   USING HELICOIL INSTALLATION TOOL 551-02.
   REMOVE TANG USING HELICOIL TANG-BREAKOFF TOOL 3695-02.

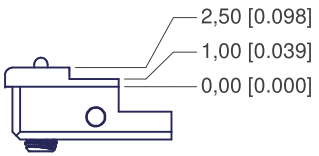
AMSAT
FOX

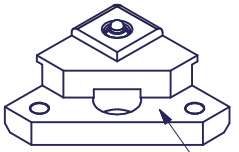
RAIL ENDS

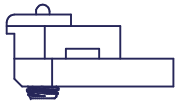
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-07-22 | FOX-ME-121 | 2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | SHEET 1 OF 6 | |

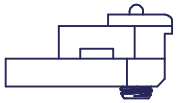


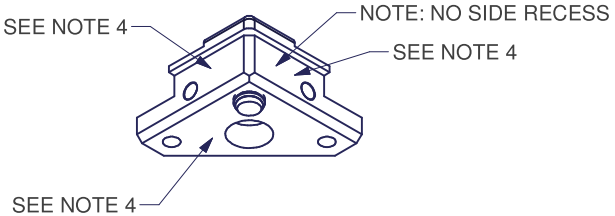


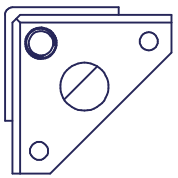






# -03 -Z W/SPRING

MAKE SIMILAR TO -01 EXCEPT AS NOTED

| ITEM | QTY | PART NUMBER | DESCRIPTION |
|---|---|---|---|
| 1 | 1 | FOX-ME-121-03 | RAIL END, -Z W/SPRING |
| 2 | 1 | MCMASTER 84985A76 | SPRING PLUNGER W/LOCK, #8-36 LONG |

Parts List

NOTES:
6) INSTALL MCMASTER-CARR 84985A76 SPRING PLUNGER W/LOCK ELEMENT, USING SPANNER WRENCH 3382A12. SPRING HOUSING FLUSH WITH ALUMINUM; BUTTON PROTRUDES ABOVE TOP, AND TAIL PROTRUDES OUT BOTTOM.

| RAIL ENDS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-22 | DWG.NO. FOX-ME-121 | REV. 2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 6 |

244

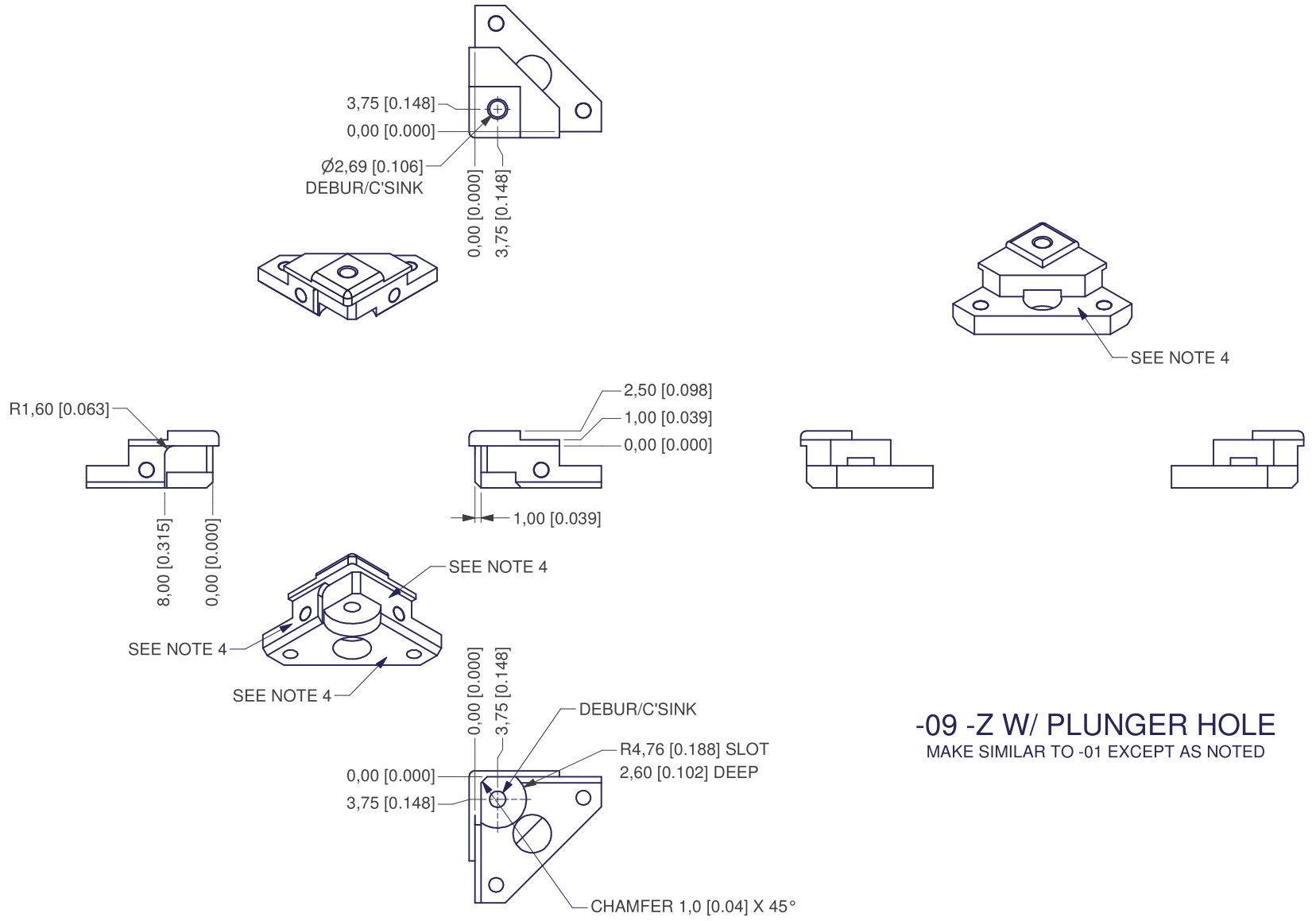
-05 -Z W/SWITCH
DELETED

| RAIL ENDS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-22 | DWG.NO. FOX-ME-121 | REV. 2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 6 |

245

-07 -Z BLANK
DELETED

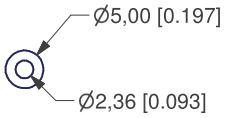
| RAIL ENDS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-22 | DWG.NO. FOX-ME-121 | REV. 2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 4 OF 6 |

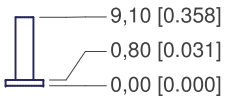



# -09 -Z W/ PLUNGER HOLE

MAKE SIMILAR TO -01 EXCEPT AS NOTED

| RAIL ENDS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-22 | DWG.NO. FOX-ME-121 | REV. 2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 5 OF 6 |

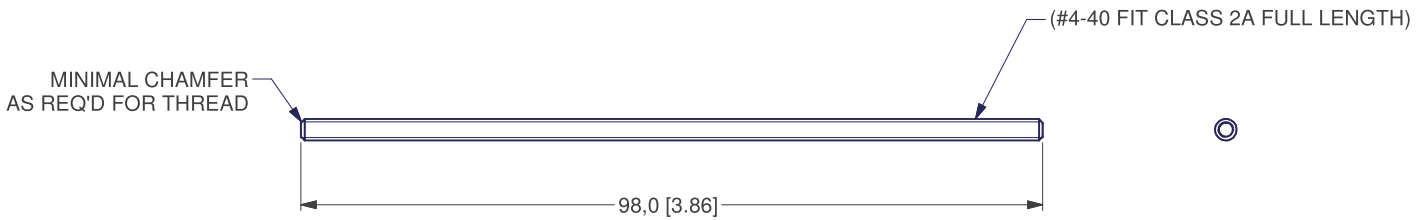


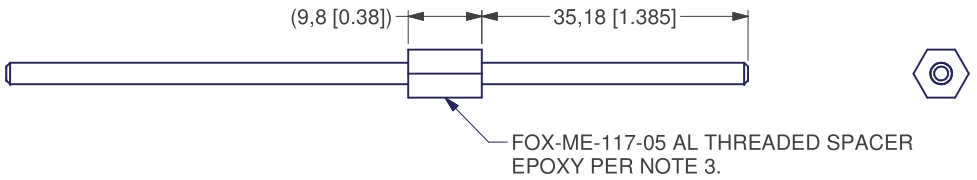




# -11 SWITCH PLUNGER

NOTES:
7) -11 MATERIAL: BLACK DELRIN

| RAIL ENDS | | | |
|---|---|---|---|
| SIZE A | DATE 2014-07-22 | DWG.NO. FOX-ME-121 | REV. 2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 6 OF 6 |



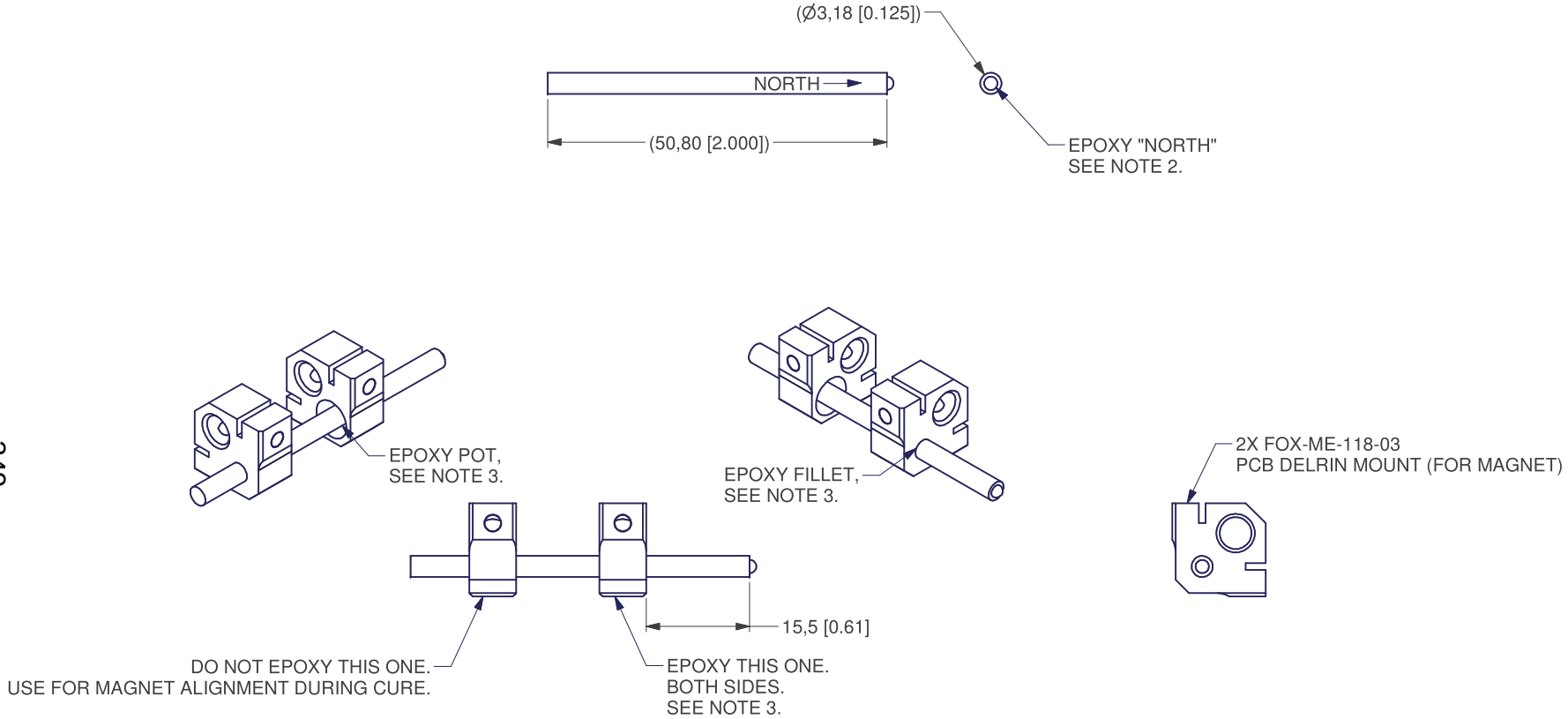

NOTES:
1) MATERIAL: 316 STAINLESS STEEL
2) MAKE FROM GRAINGER 21YN73 OR EQUIVALENT.
3) DIP PARTS IN IPA AND WIPE THREADS. AIR DRY.
   APPLY HYSOL/LOCTITE EPOXI-PATCH 9340 TO ROD THREADS. CURE PER TDS.

AMSAT
FOX

THREADED ROD

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2016-06-17 | FOX-ME-123 | 0.3 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 1 |




NOTES:
1) SOURCE: K&J MAGNETICS. P/N: D2Y0. DESCRIPTION: 1/8"X2" CYLINDER, N42, NI.
2) TO NOTE "COMPASS NORTH" END, DOT EPOXY OF HYSOL/LOCTITE EPOXI-PATCH 9340.
3) EPOXY MAGNET INTO ONLY ONE STANDOFF, USING HYSOL/LOCTITE EPOXI-PATCH 9340.
   USE ADDITIONAL MOUNT WITHOUT EPOXY TO MAINTAIN PROPER ALIGNMENT OF MAGNET DURING CURE.
   CURE PER TDS.

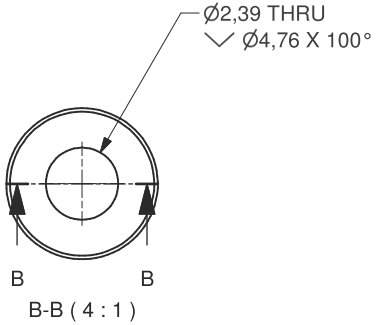
AMSAT
FOX

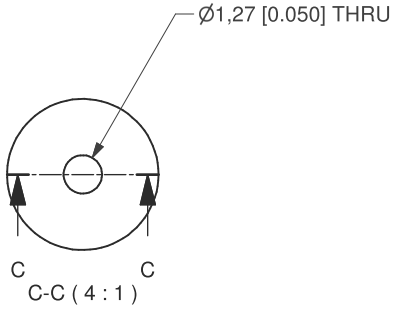
MAGNET

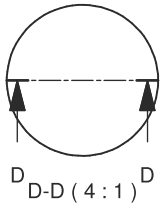
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2016-06-17 | FOX-ME-124 | 0.2 |

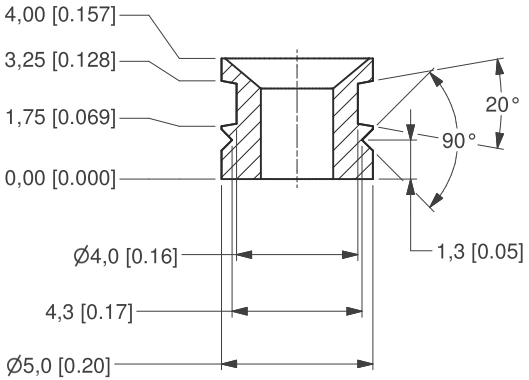
| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 1 |

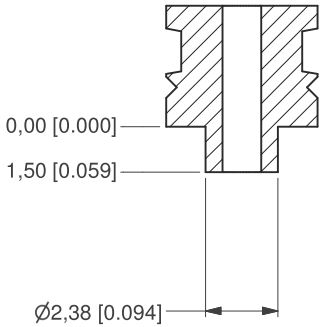
250

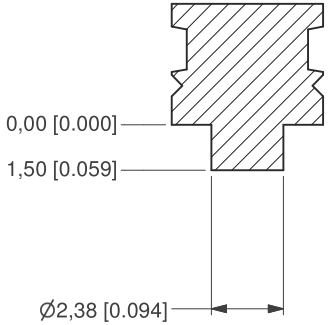



## -01
DELETED

## -03
BLACK DELRIN,
MCMASTER-CARR 8576K11

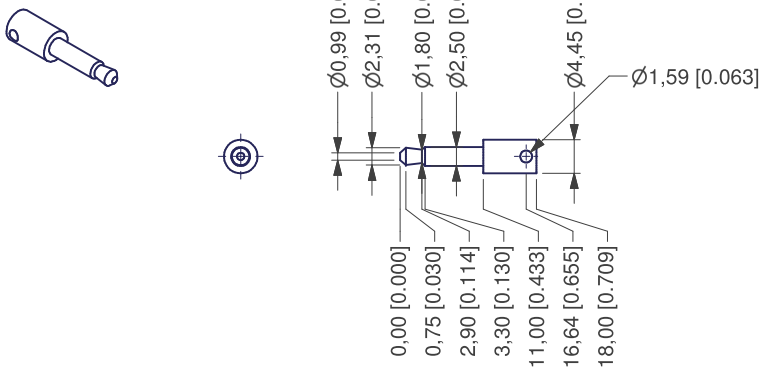
## -05
DIMS SAME AS -03 EXCEPT AS NOTED.
COPPER ALLOY 110 H04 (HARD),
MCMASTER-CARR 8966K423.
GOLD PLATED.

## -07
DIMS SAME AS -03 EXCEPT AS NOTED.
G10/FR4 ROD,
MCMASTER-CARR 8669K891
OR EQUIVALENT.

AMSAT
FOX

ANTENNA POST

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-04-19 | FOX-ME-126 | 0.5 |
| SCALE 4 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | SHEET 1 OF 1 | |



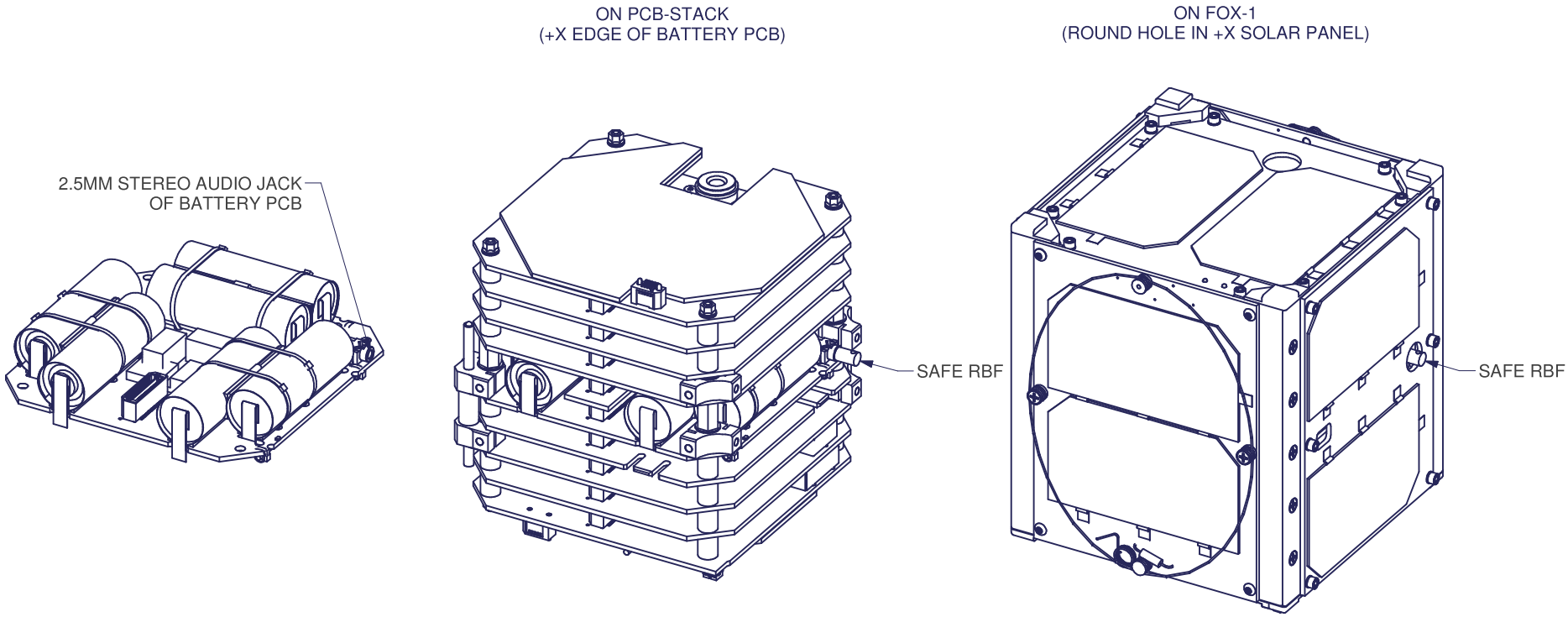

NOTES:
1) MATERIAL: NICKEL ALLOY 200/201
2) NOT SHOWN: A FINGER-SIZED LOOP OF RED WIRE OR BREYDEN 204-3 LOG LACING CORD/TAPE MAY BE USED THRU HOLE.

AMSAT
FOX

SAFE RBF

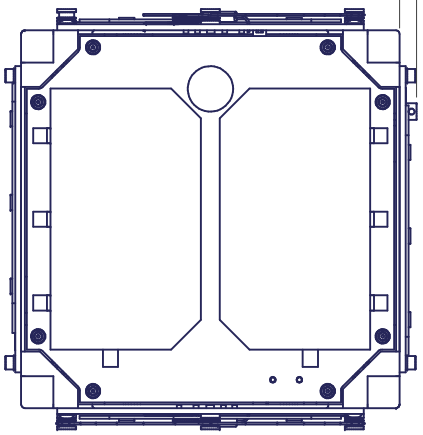
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-04-24 | FOX-ME-130 | 0.2 |

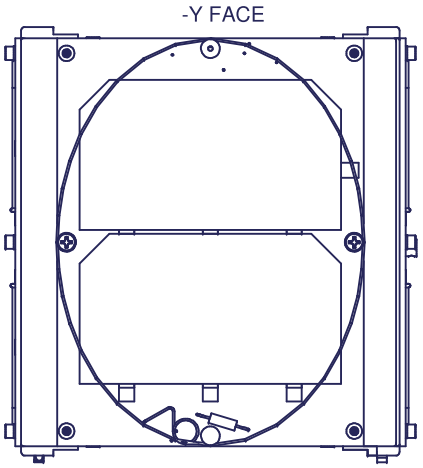
| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 4 |

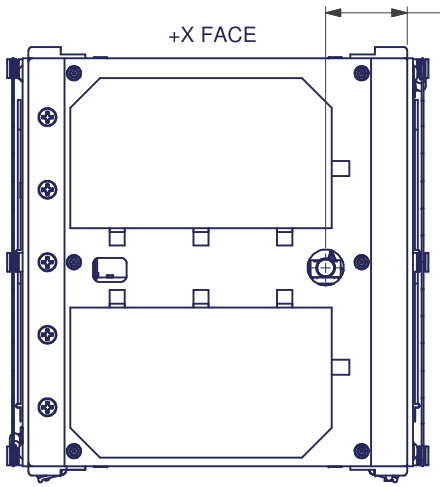



NOTES:
3) TO SAFE (DISABLE) ELECTRONICS:
FULLY INSERT INTO AUDIO JACK ON +X SIDE OF BATTERY BOARD
(ROUND HOLE IN +X SOLAR PANEL).
4) TO ARM (ENABLE) ELECTRONICS:
FULLY REMOVE.

| SAFE RBF | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-24 | DWG.NO. FOX-ME-130 | REV. 0.2 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 4 |

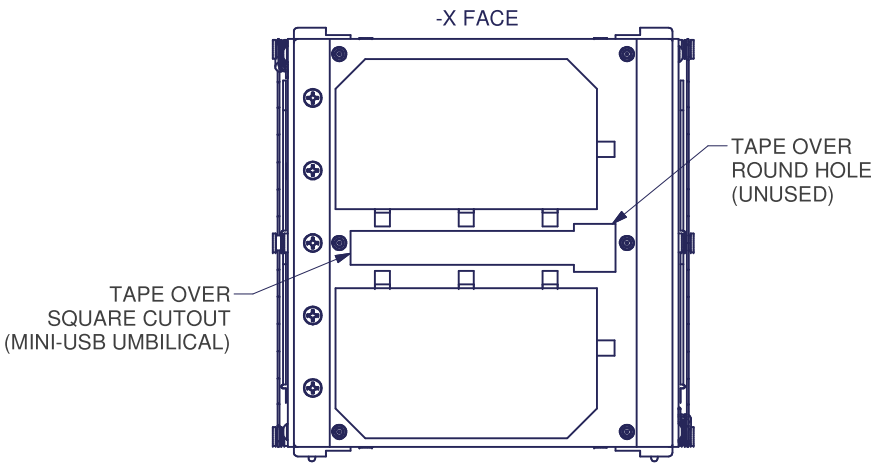


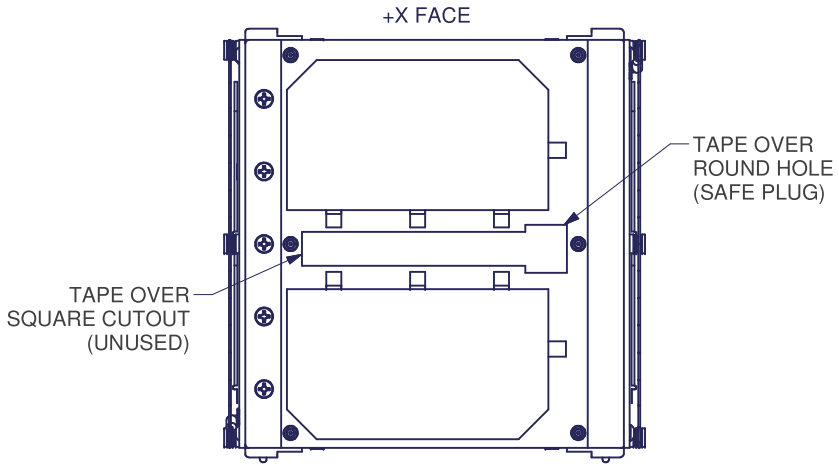



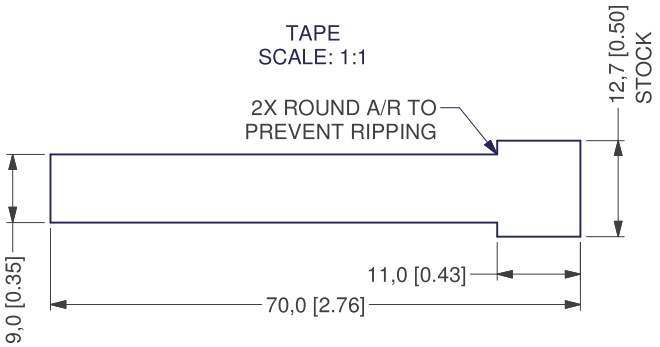




NOTES:
5) COMPATIBLE WITH P-POD ALLOWABLE ENVELOPE.
6) REMOVE BEFORE FLIGHT (RBF) ITEM.

| SAFE RBF | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-24 | DWG.NO. FOX-ME-130 | REV. 0.2 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 4 |

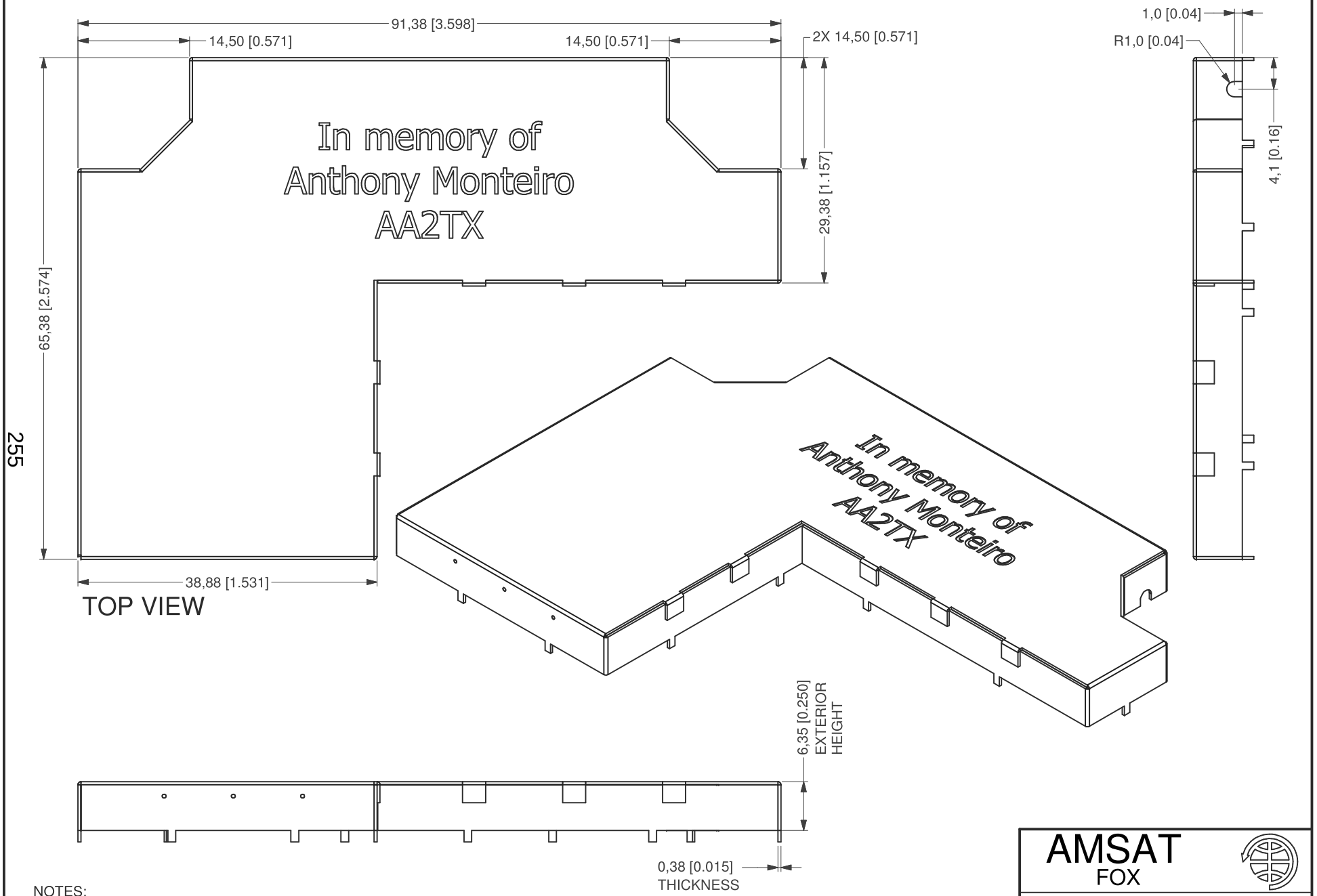



NOTES:
7) UMBILICAL HOLES TO BE COVERED FOR FLIGHT USING APPROVED 1ST SURF METAL TAPE.
UNLESS OTHERWISE NOTED, APPROVED 1ST SURFACE INCLUDES ALUMINUM FOIL W/ ACRYLIC ADHESIVE 3M 425.
FOR SAFE RBF, AND POSSIBLY ALSO THE MINI-USB, THIS MAY HAVE TO BE PERFORMED
THROUGH ACCESS PORT(S) INSIDE P-POD.
CAUTION: DO NOT SHORT SOLAR PANEL EXPOSED WITH ALUMINUM FOIL TAPE.

| SAFE RBF | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-24 | DWG.NO. FOX-ME-130 | REV. 0.2 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 4 OF 4 |




255

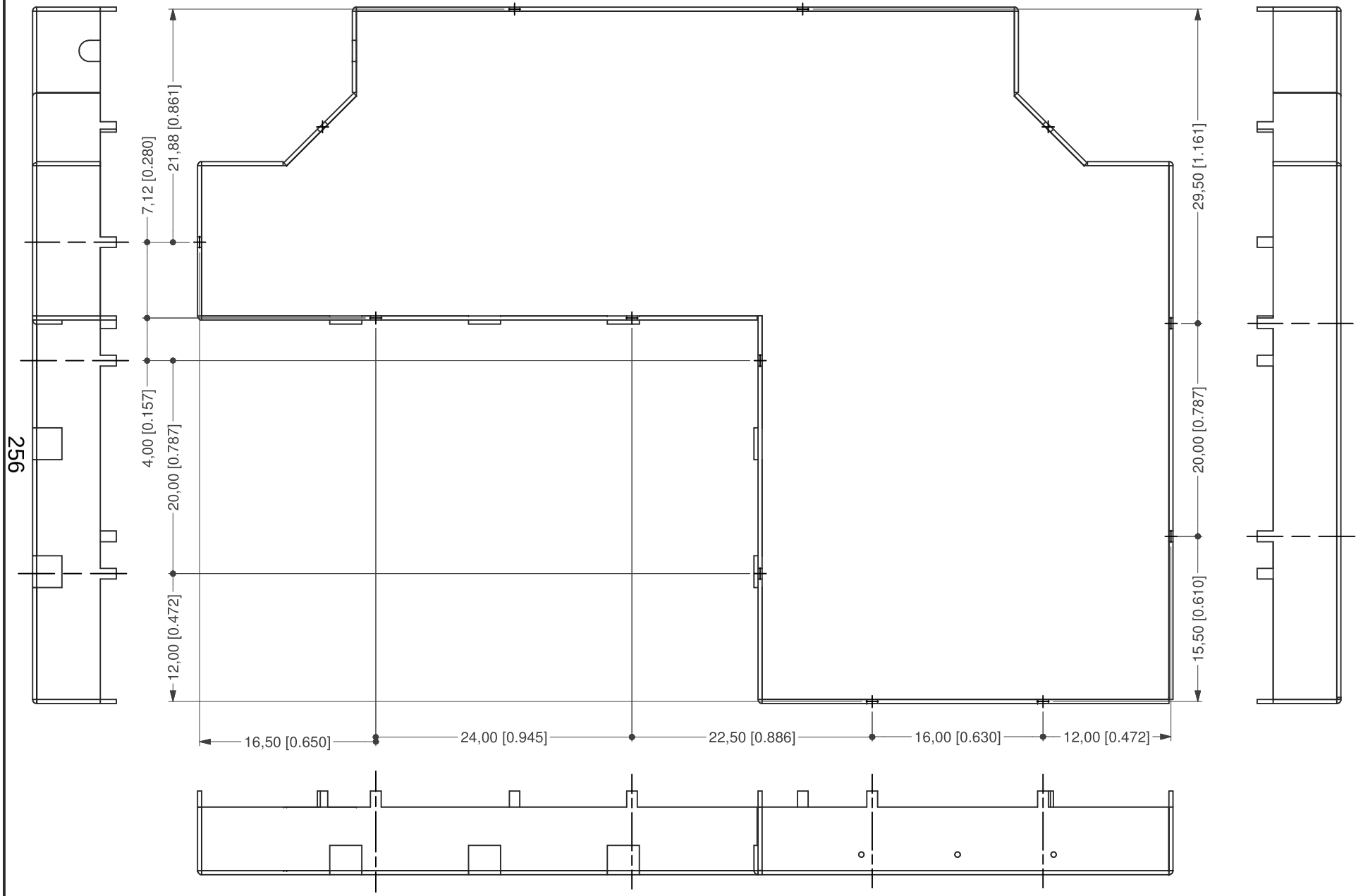
NOTES:
1) MATERIAL: PURE COPPER
FINISH: 63/37 (EUTECTIC) TIN/LEAD
2) ETCH "In memory of Anthony Monteiro AA2TX" IS NOT THRU.

AMSAT
FOX

RX RF SHIELD

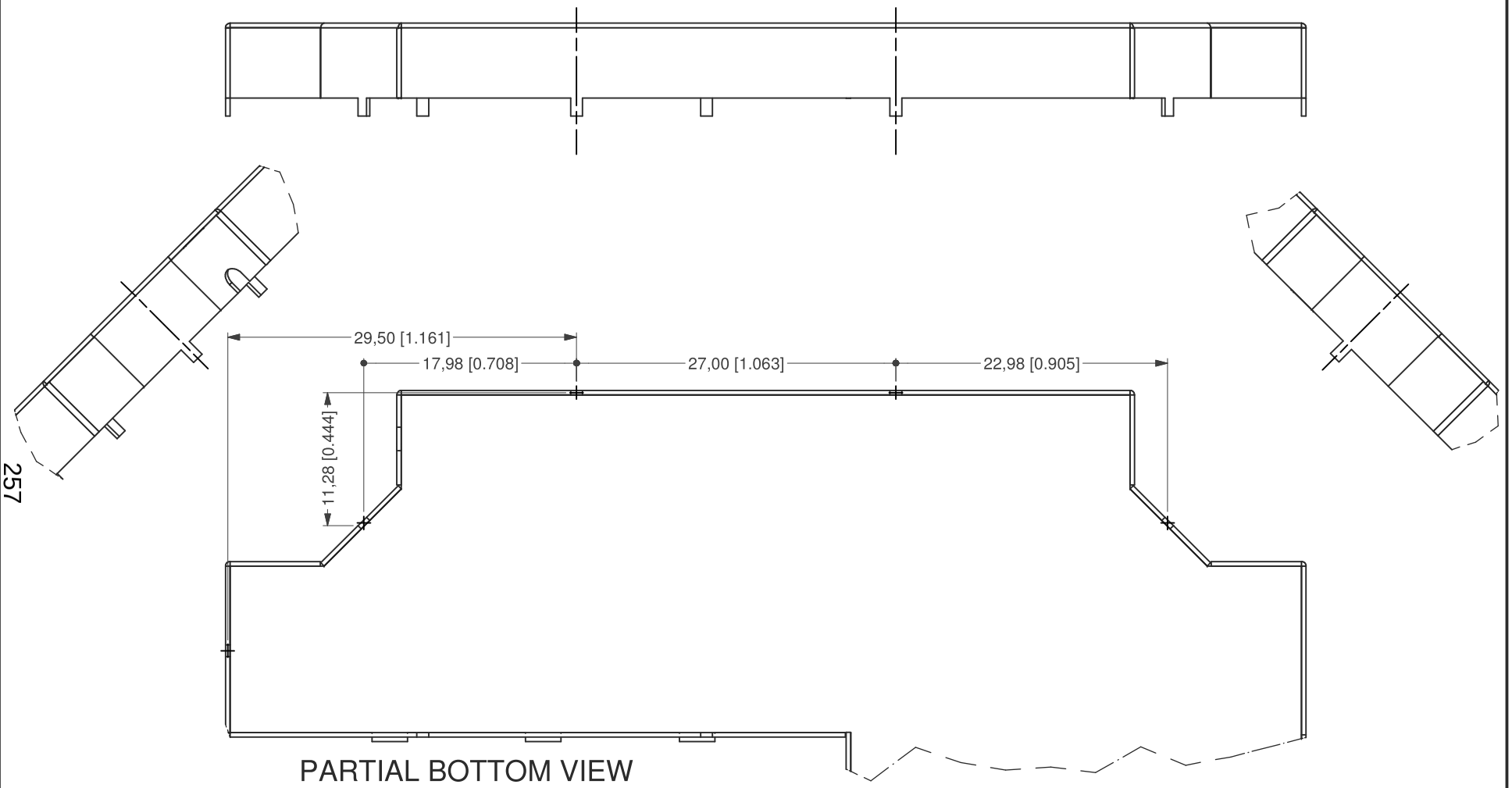
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-04-19 | FOX-ME-131 | 4 |
| SCALE 3 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | SHEET | 1 OF 4 |

## BOTTOM VIEW




NOTES:
3) TABS SHOWN ARE DIMENSIONED TO MATCH PCB PLATED HOLES.
4) RECESSED TABS (NOT SHOWN) ARE NECESSARY FOR PUNCHOUT SHEET.

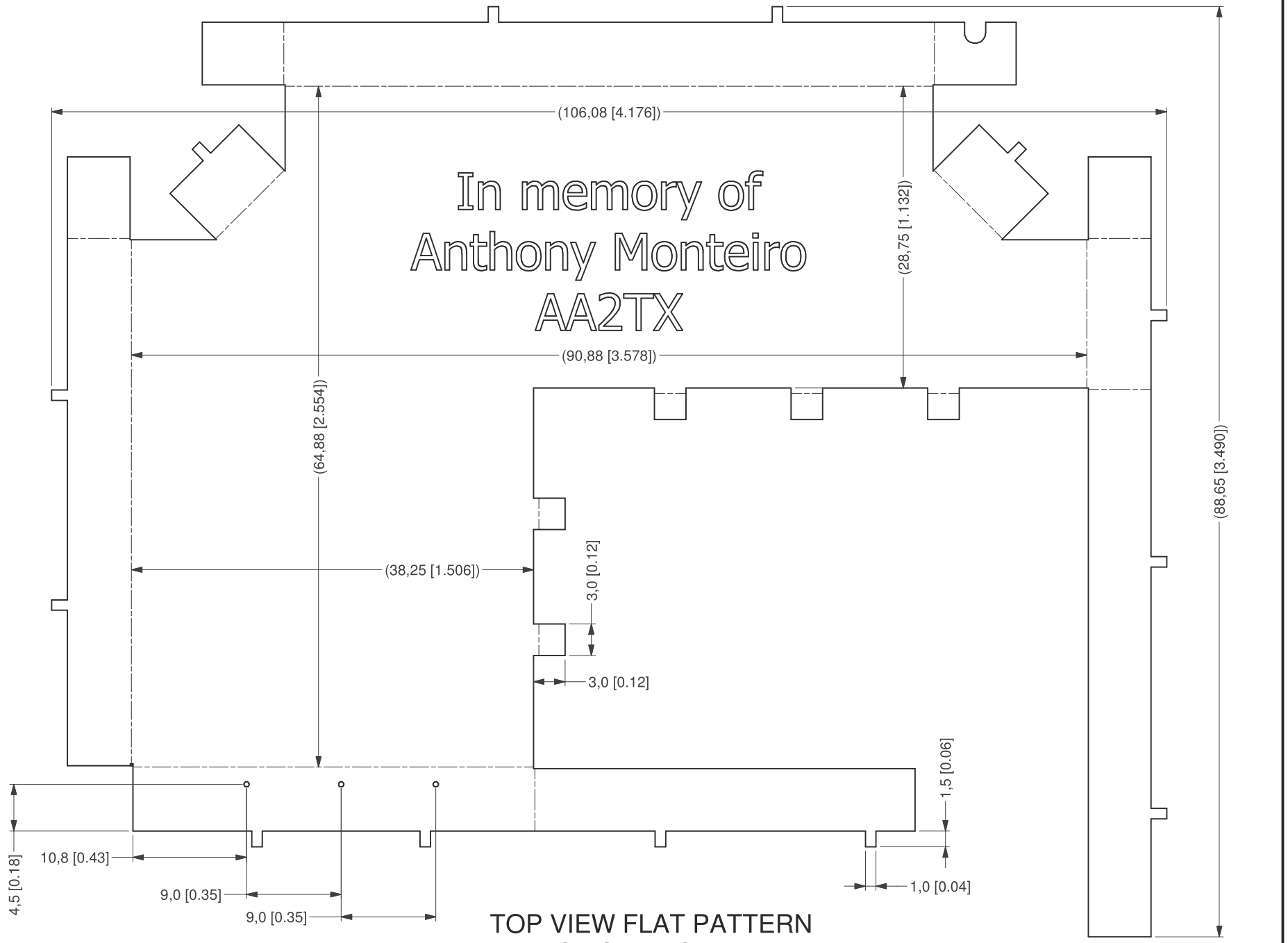
| RX RF SHIELD | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-19 | DWG.NO. FOX-ME-131 | REV. 4 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 4 |




PARTIAL BOTTOM VIEW

257

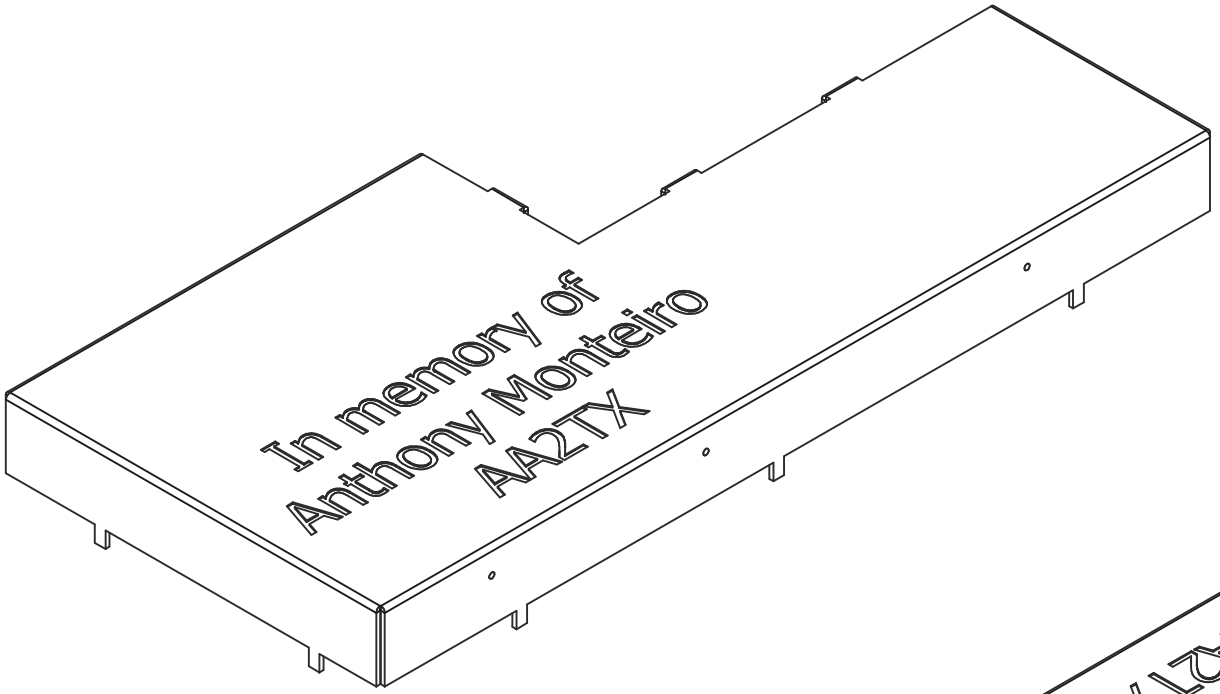
| RX RF SHIELD | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-19 | DWG.NO. FOX-ME-131 | REV. 4 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 of 4 |

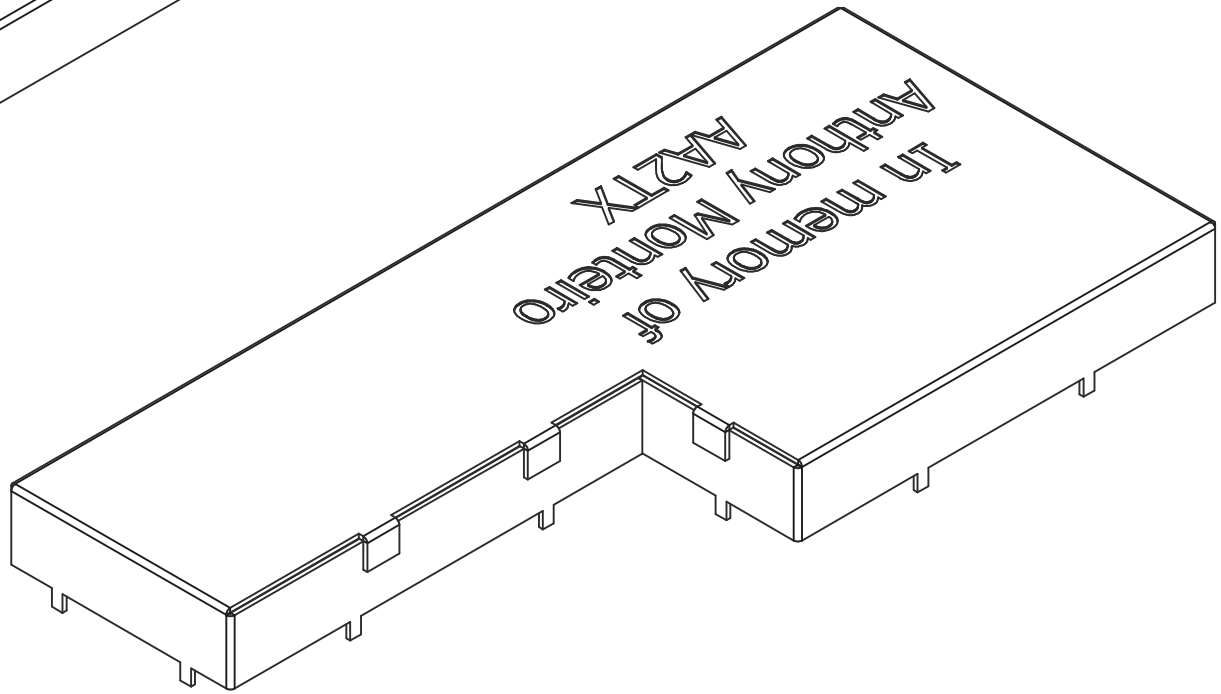



TOP VIEW FLAT PATTERN
ALL FLAPS FOLD DOWNWARD 90°

DIMENSIONS ARE REFERENCE
SEE BENT PATTERN DIMENSIONS

| RX RF SHIELD | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-19 | DWG.NO. FOX-ME-131 | REV. 4 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 4 of 4 |

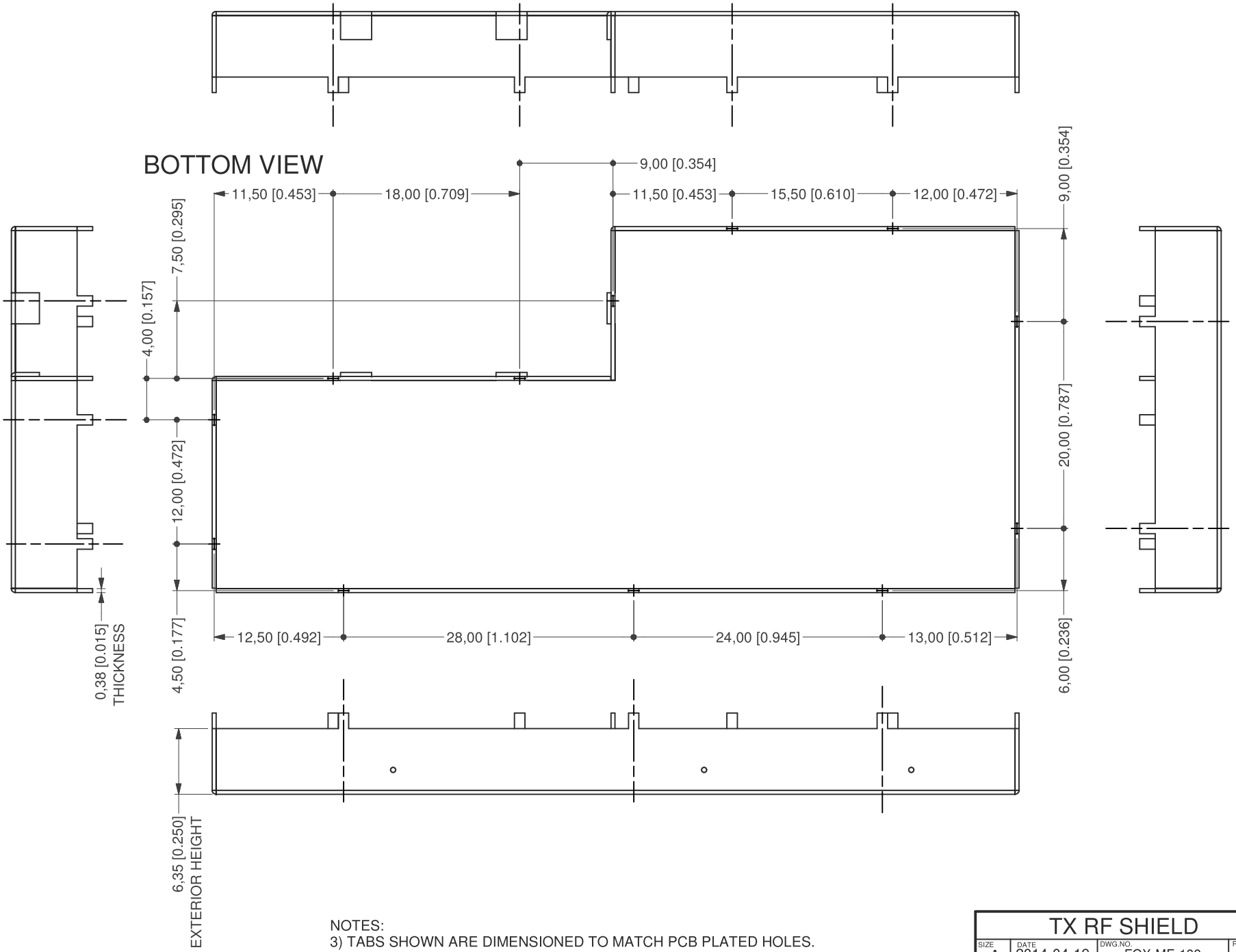



259

NOTES:
1) MATERIAL: PURE COPPER
FINISH: 63/37 (EUTECTIC) TIN/LEAD
2) ETCHING "In memory of Anthony Monteiro AA2TX" IS NOT THRU.

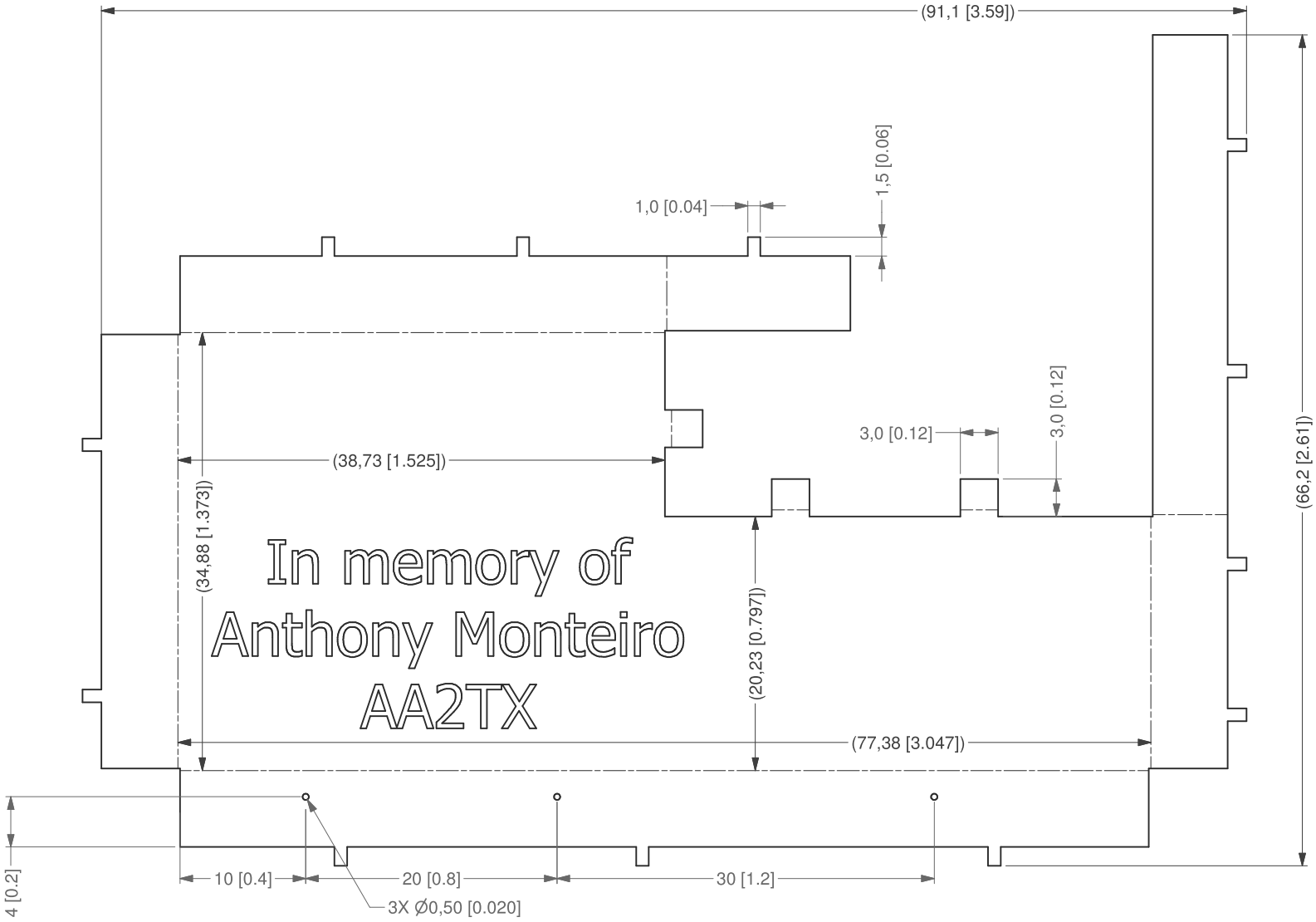
| AMSAT FOX | | | |
|---|---|---|---|
| TX RF SHIELD | | | |
| SIZE A | DATE 2014-04-19 | DWG.NO. FOX-ME-132 | REV. 2 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | | SHEET 1 OF 3 |




260

NOTES:
3) TABS SHOWN ARE DIMENSIONED TO MATCH PCB PLATED HOLES.
4) RECESSED TABS (NOT SHOWN) AS NECESSARY FOR PUNCHOUT SHEET.

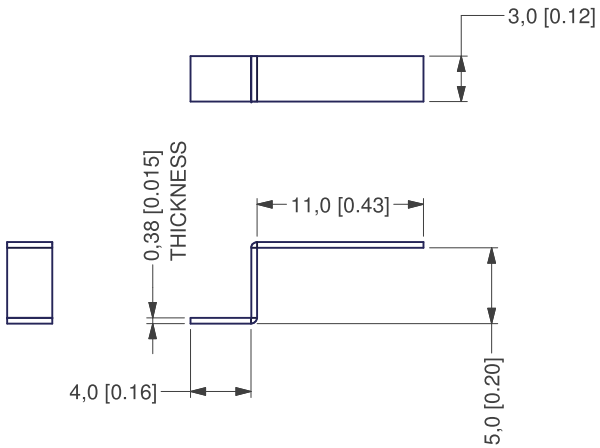
| TX RF SHIELD | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-19 | DWG.NO. FOX-ME-132 | REV. 2 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 3 |




## TOP VIEW FLAT PATTERN
## ALL FLAPS FOLD DOWNWARD 90°

DIMENSIONS ARE REFERENCE
SEE BENT PATTERN DIMENSIONS

| TX RF SHIELD | | | |
|---|---|---|---|
| SIZE A | DATE 2014-04-19 | DWG.NO. FOX-ME-132 | REV. 2 |
| SCALE 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 3 |



FLAT PATTERN

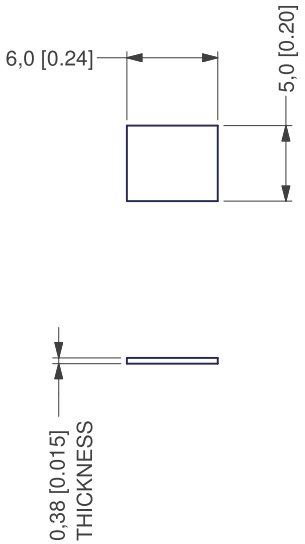
NOTES:
1) MATERIAL: PURE COPPER
   FINISH: 63/37 (EUTECTIC) TIN/LEAD
2) TABS (NOT SHOWN) AS NECESSARY FOR PUNCHOUT SHEET.

AMSAT
FOX

CRYSTAL L STRIP

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-04-18 | FOX-ME-133 | 1 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 2 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 1 |



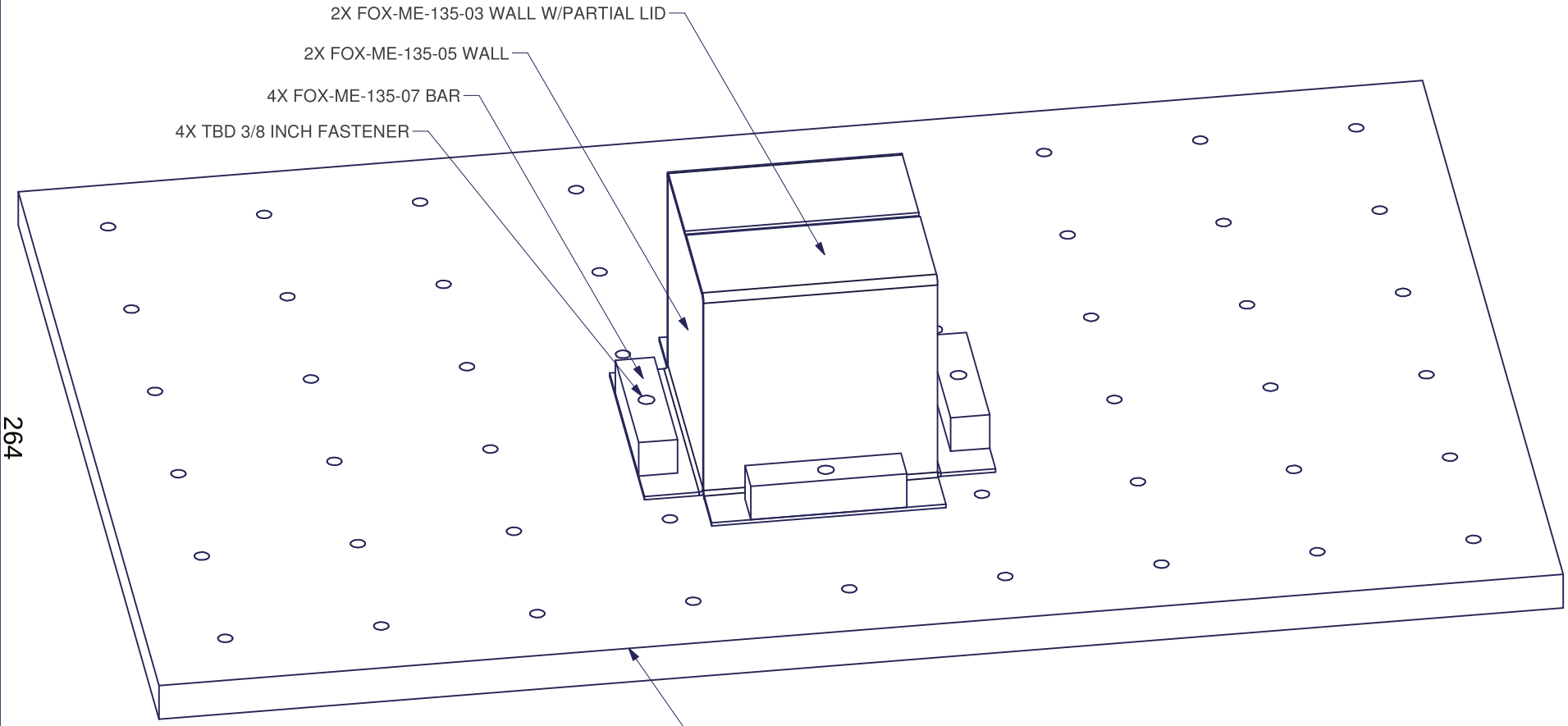

NOTES:
1) MATERIAL: PURE COPPER
   FINISH: 63/37 (EUTECTIC) TIN/LEAD
2) TABS (NOT SHOWN) AS NECESSARY FOR PUNCHOUT SHEET.

AMSAT
FOX

RX SHIM

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-04-18 | FOX-ME-134 | 1 |
| SCALE | 2 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | SHEET 1 OF 1 |

# -01 HAT ASSEMBLY AT THERMAL VACUUM BAKEOUT





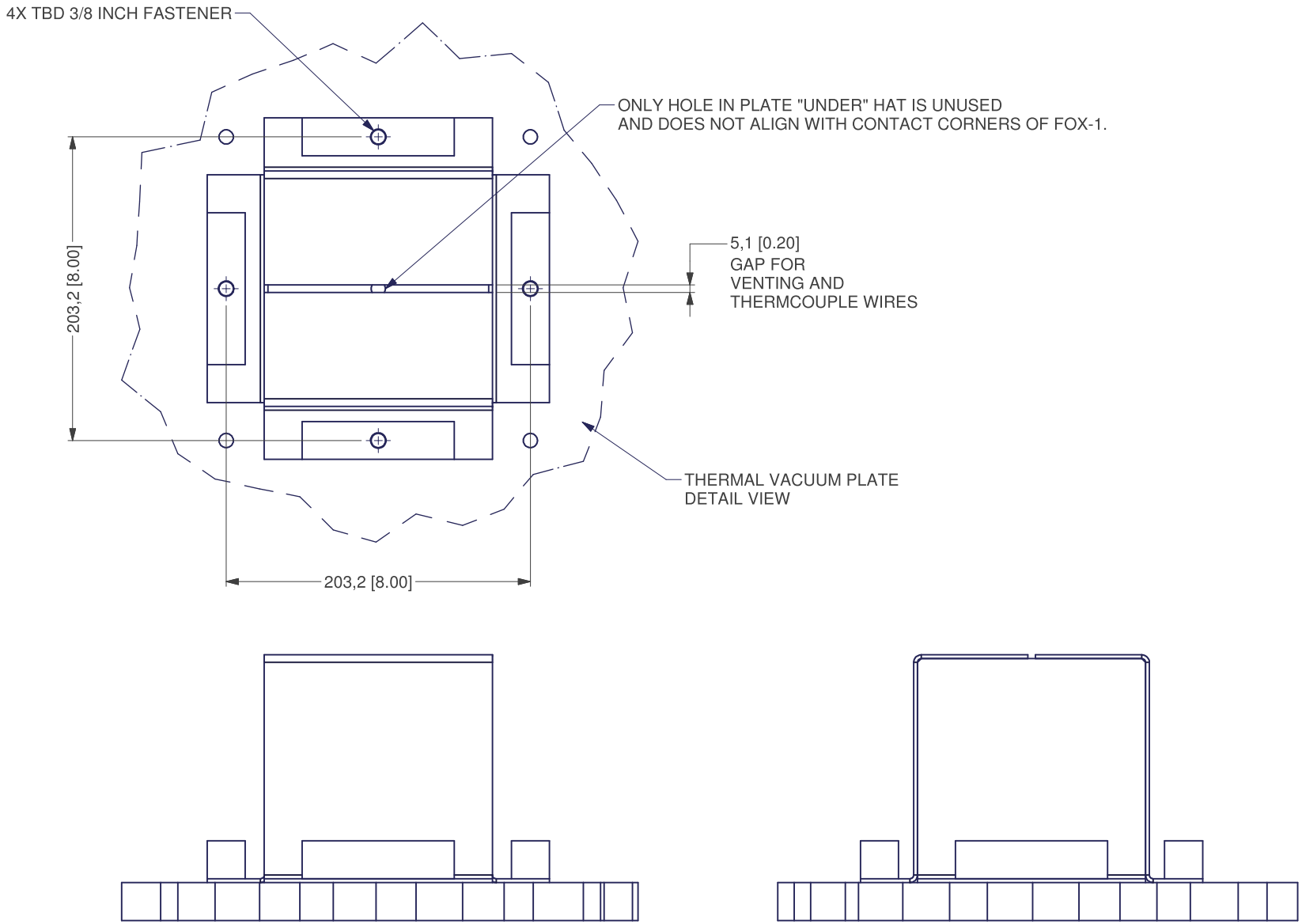

NOTES:
1) HAT MAY BE LOCATED AND ORTHOGONALLY ROTATED AS NECESSARY.

AMSAT
FOX

HAT, BAKEOUT

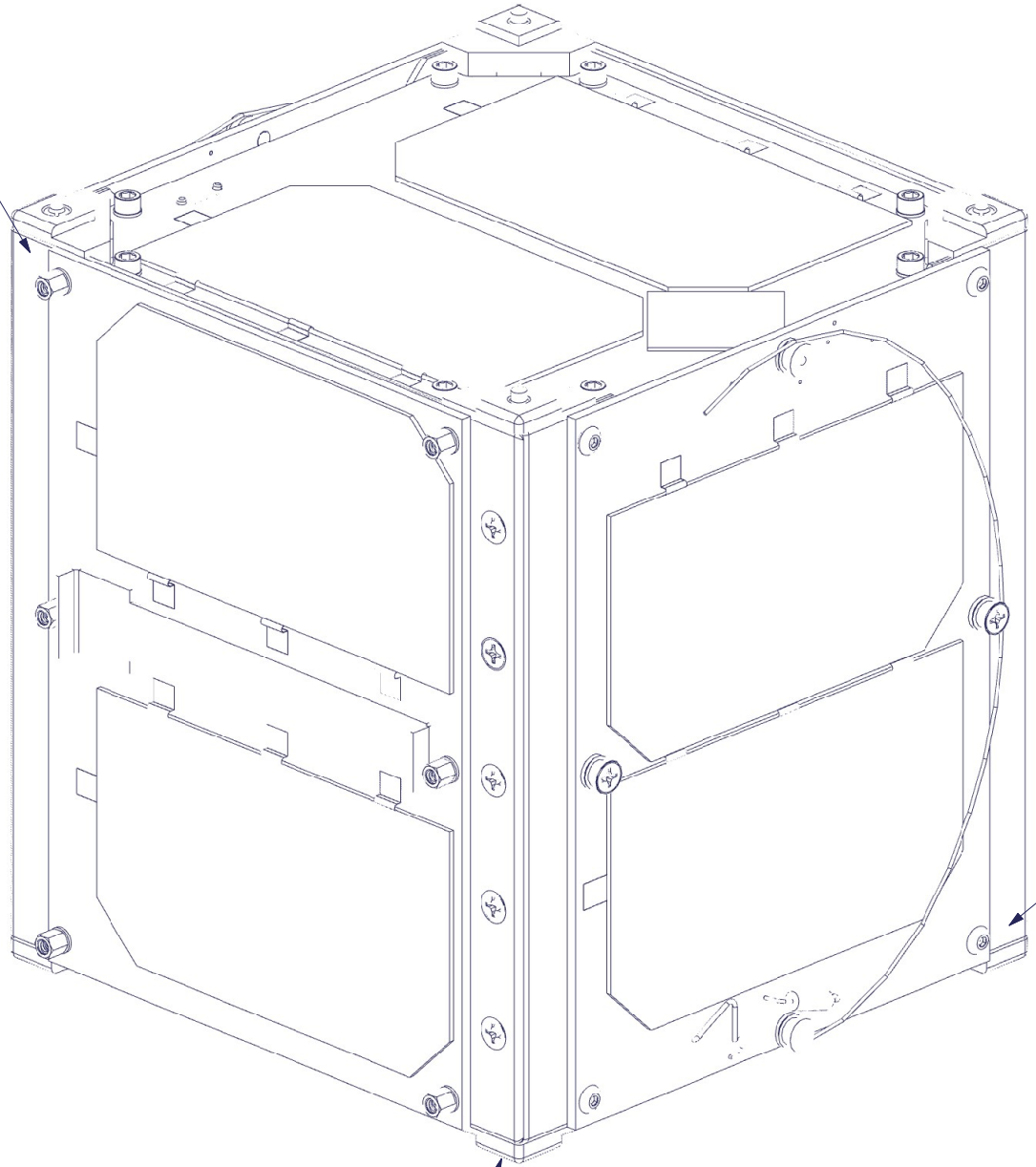
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-09-18 | FOX-ME-135 | 0.3 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 4 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 6 |





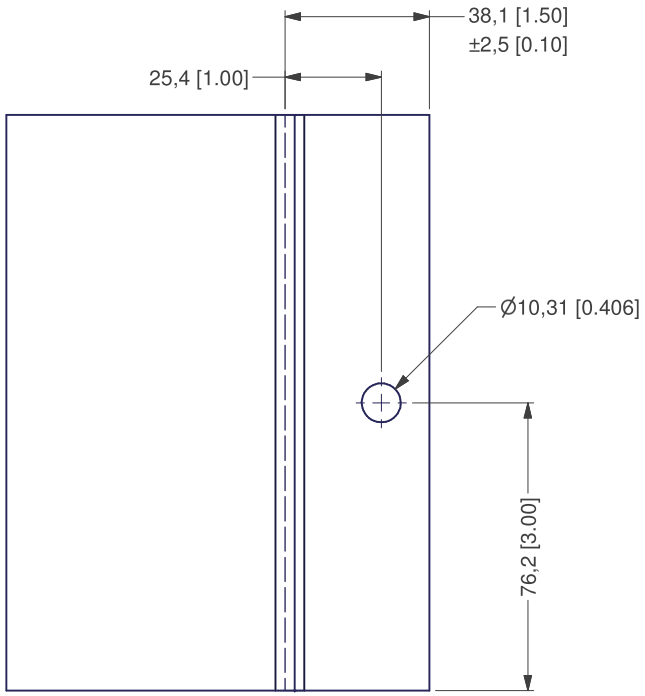

| HAT, BAKEOUT | | | |
|---|---|---|---|
| SIZE A | DATE 2014-09-18 | DWG.NO. FOX-ME-135 | REV. 0.3 |
| SCALE 1 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 6 |



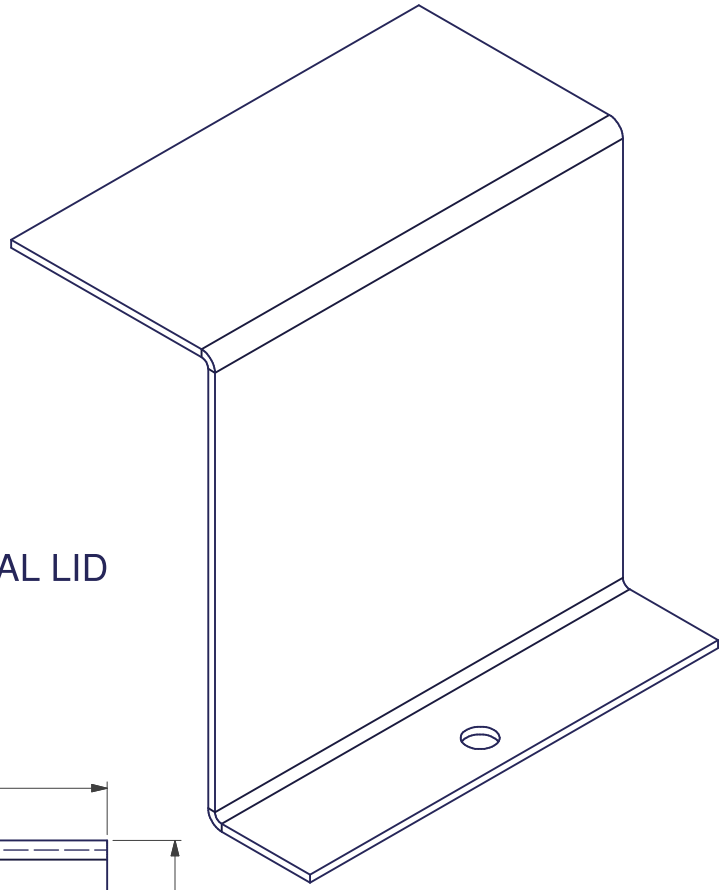

FOX-1 RESTS ON FOUR FLAT +Z CORNERS WITHOUT FASTENERS.
ORTHOGONAL ROTATION OF FOX-1 AS NECESSARY.
PLATE TO BE FLAT AND CLEAR OF HOLES/PROJECTS/DEBRIS AT CONTACT POINTS.

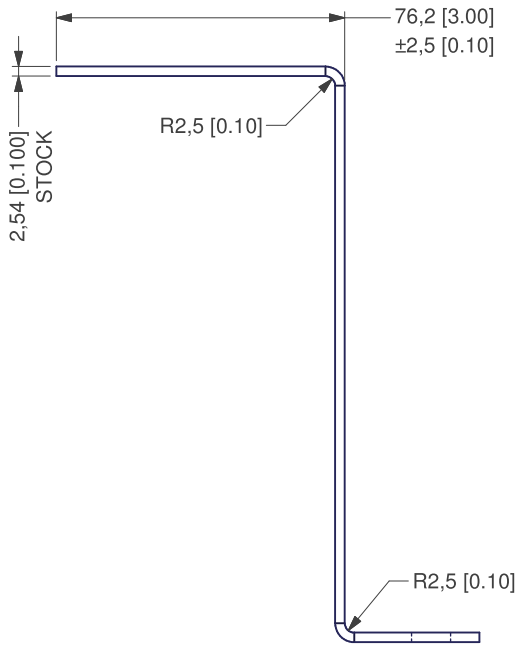
# HAT, BAKEOUT

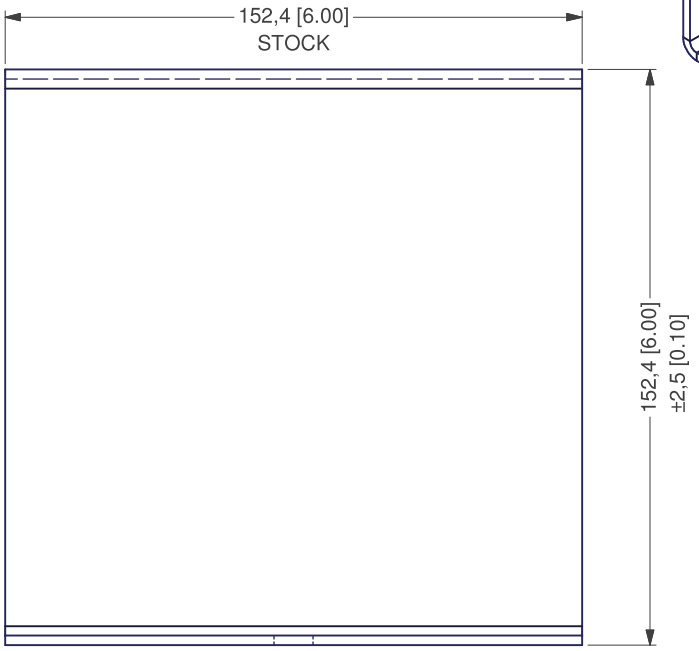
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-09-18 | FOX-ME-135 | 0.3 |
| SCALE | 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | SHEET 3 OF 6 |




# -03 WALL WITH PARTIAL LID

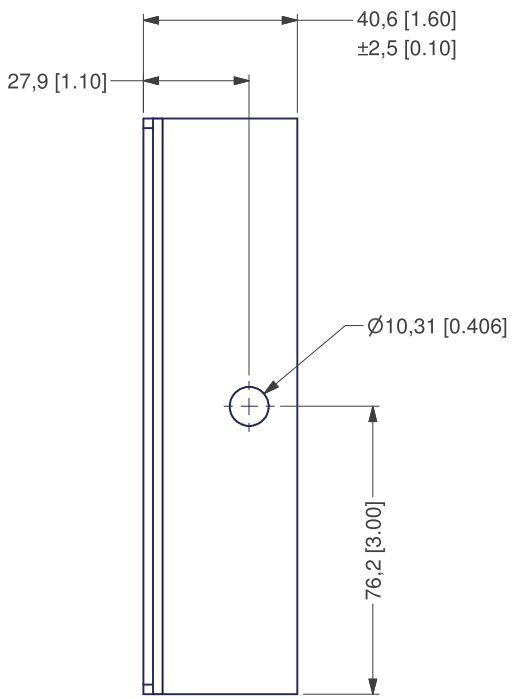




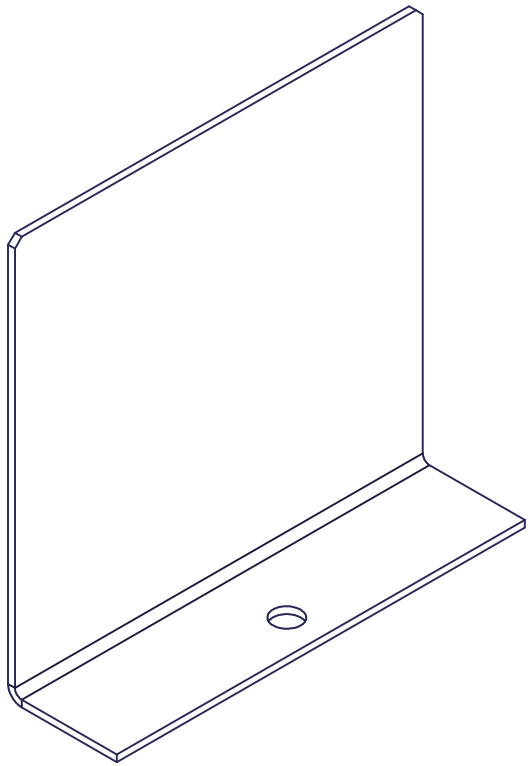



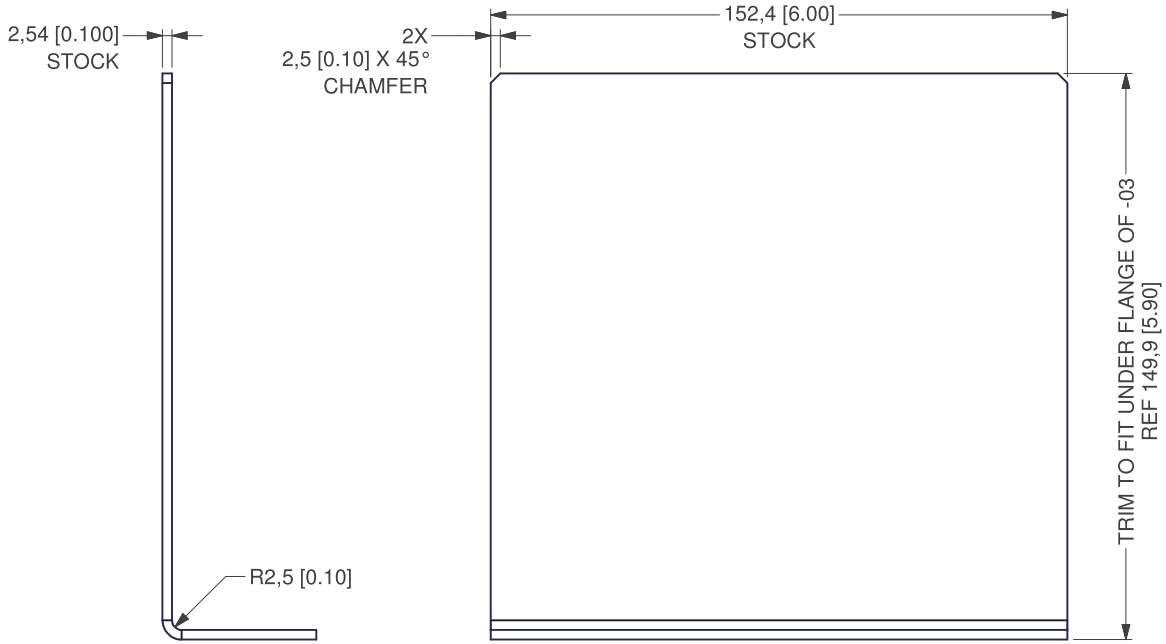

NOTES:
2) MATERIAL: ALUMINUM 5052 H32
MAKE FROM MCMASTER-CARR 88895K28
3) FINISH: BLACK ANODIZE

| HAT, BAKEOUT | | | |
|---|---|---|---|
| SIZE A | DATE 2014-09-18 | DWG.NO. FOX-ME-135 | REV. 0.3 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 4 OF 6 |




## -05 WALL

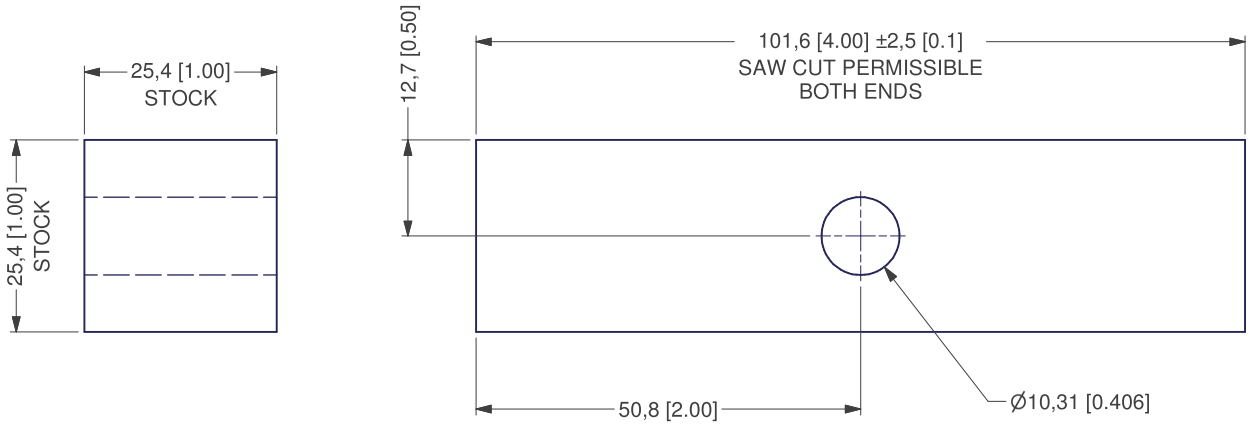





NOTES:
4) MATERIAL: ALUMINUM 5052 H32
MAKE FROM MCMASTER-CARR 88895K28
5) FINISH: BLACK ANODIZE

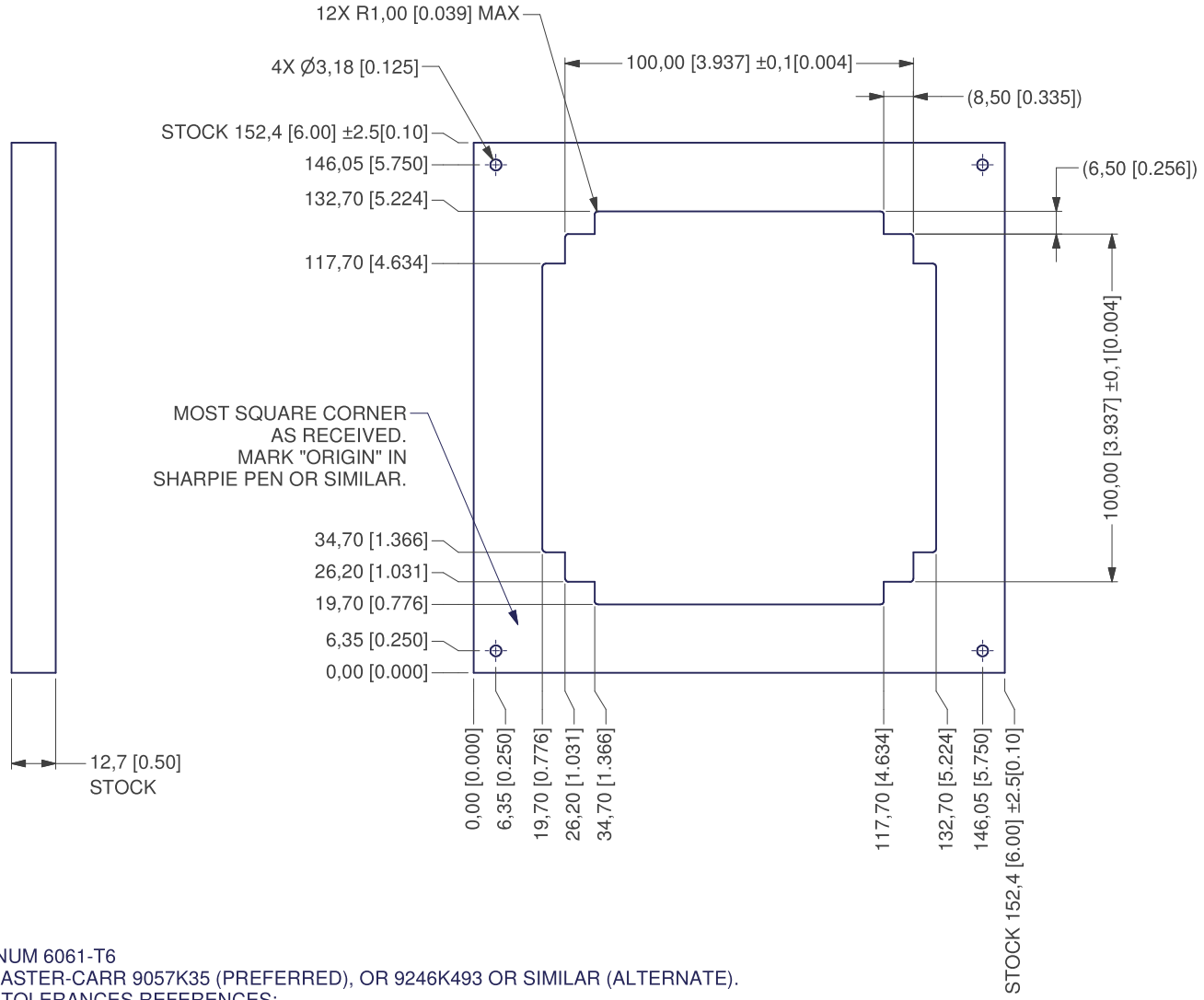
| HAT, BAKEOUT | | | |
|---|---|---|---|
| SIZE A | DATE 2014-09-18 | DWG.NO. FOX-ME-135 | REV. 0.3 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 5 OF 6 |

# -07 BAR




NOTES:
6) MATERIAL: 1215 CARBON STEEL OR SIMILAR
MAKE FROM MCMASTER-CARR 4416T47 OR SIMILAR.
7) TBD SHIM ENDS MAX 0,25 [0.010] MAX TO SPREAD OUT COMPRESSIVE FORCE.

| HAT, BAKEOUT | | | |
|---|---|---|---|
| SIZE A | DATE 2014-09-18 | DWG.NO. FOX-ME-135 | REV. 0.3 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 6 OF 6 |




NOTES:
1) MATERIAL: ALUMINUM 6061-T6
   MAKE FROM MCMASTER-CARR 9057K35 (PREFERRED), OR 9246K493 OR SIMILAR (ALTERNATE).
2) DIMENSIONS AND TOLERANCES REFERENCES:
   *CUBESAT DESIGN SPECIFICATION REV 13* BY CALIFORNIA POLYTECHNIC STATE UNIVERSITY
   SECTION 3.2.3
   APPENDIX B, SECTION 1, 1U CUBESAT DESIGN SPECIFICATION DRAWING
   APPENDIX C, SECTION 1, 1U CUBESAT ACCEPTANCE CHECKLIST

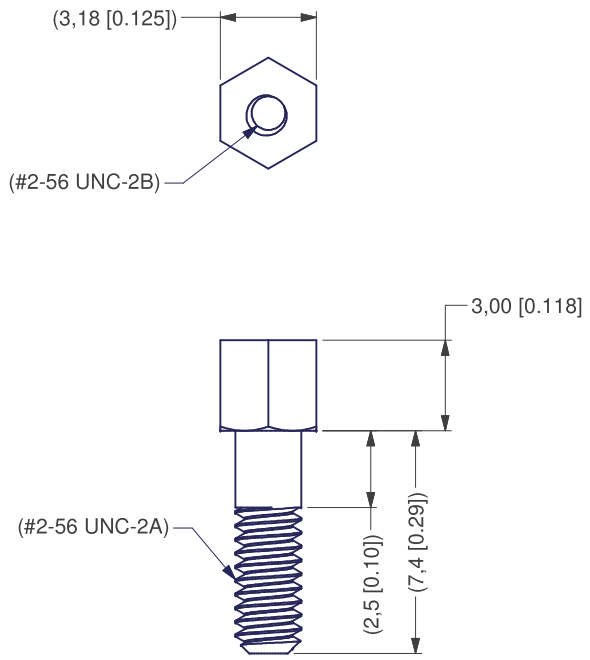
AMSAT
FOX

PPOD CROSS-SECTION

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2014-09-11 | FOX-ME-137 | 0.1 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 1 |

270



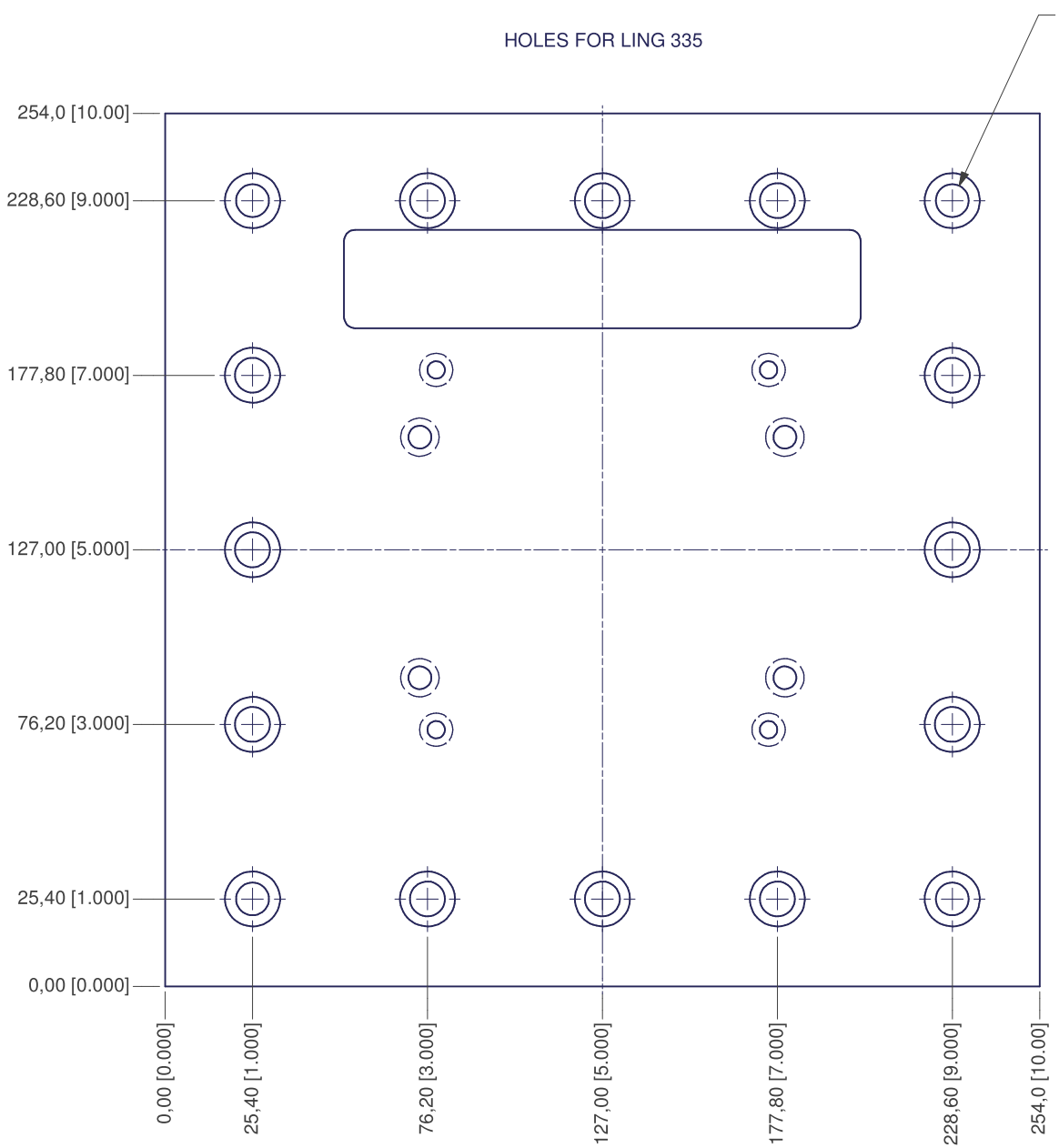

NOTES:
1) MAKE FROM GLENAIR 080-00-00-100 (83513/05-07) JACKPOST KIT.
REF MATERIAL: 300 SERIES STAINLESS STEEL.

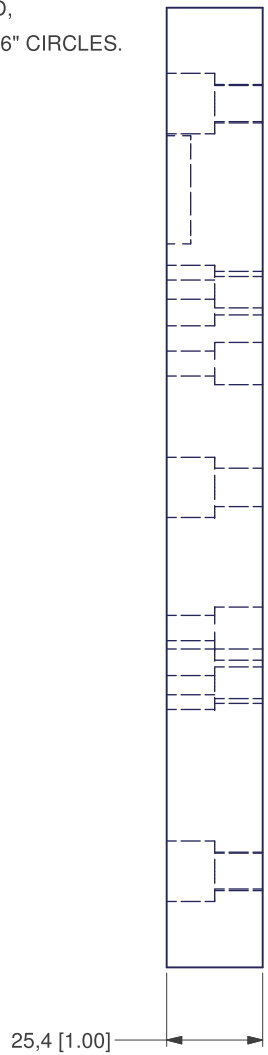
AMSAT
FOX

JACKPOST

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2016-01-21 | FOX-ME-138 | 0.1 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 4 : 1 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 1 |



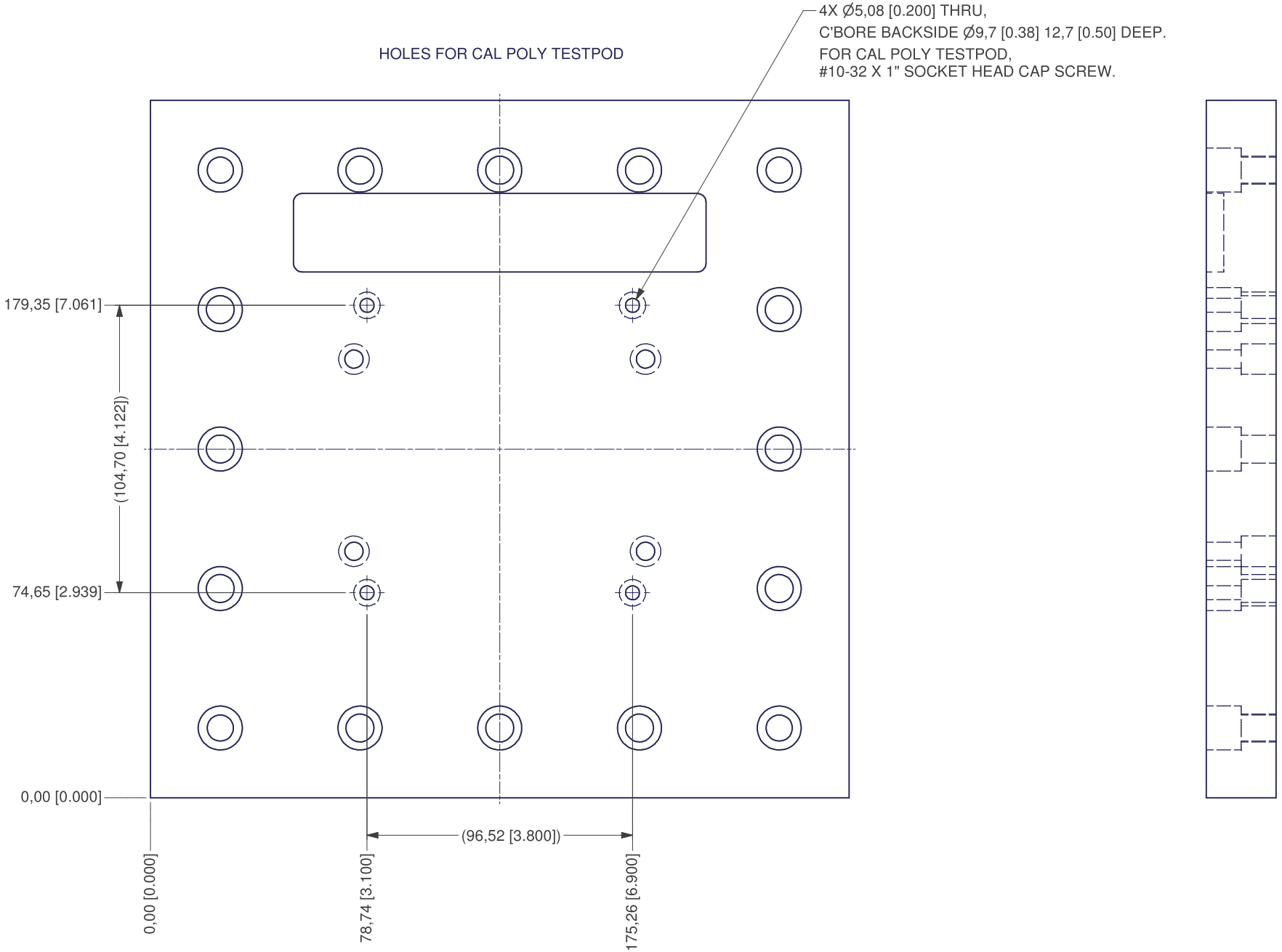

NOTES:
1) MATERIAL: ALUMINUM 6061-T6

272

AMSAT
FOX

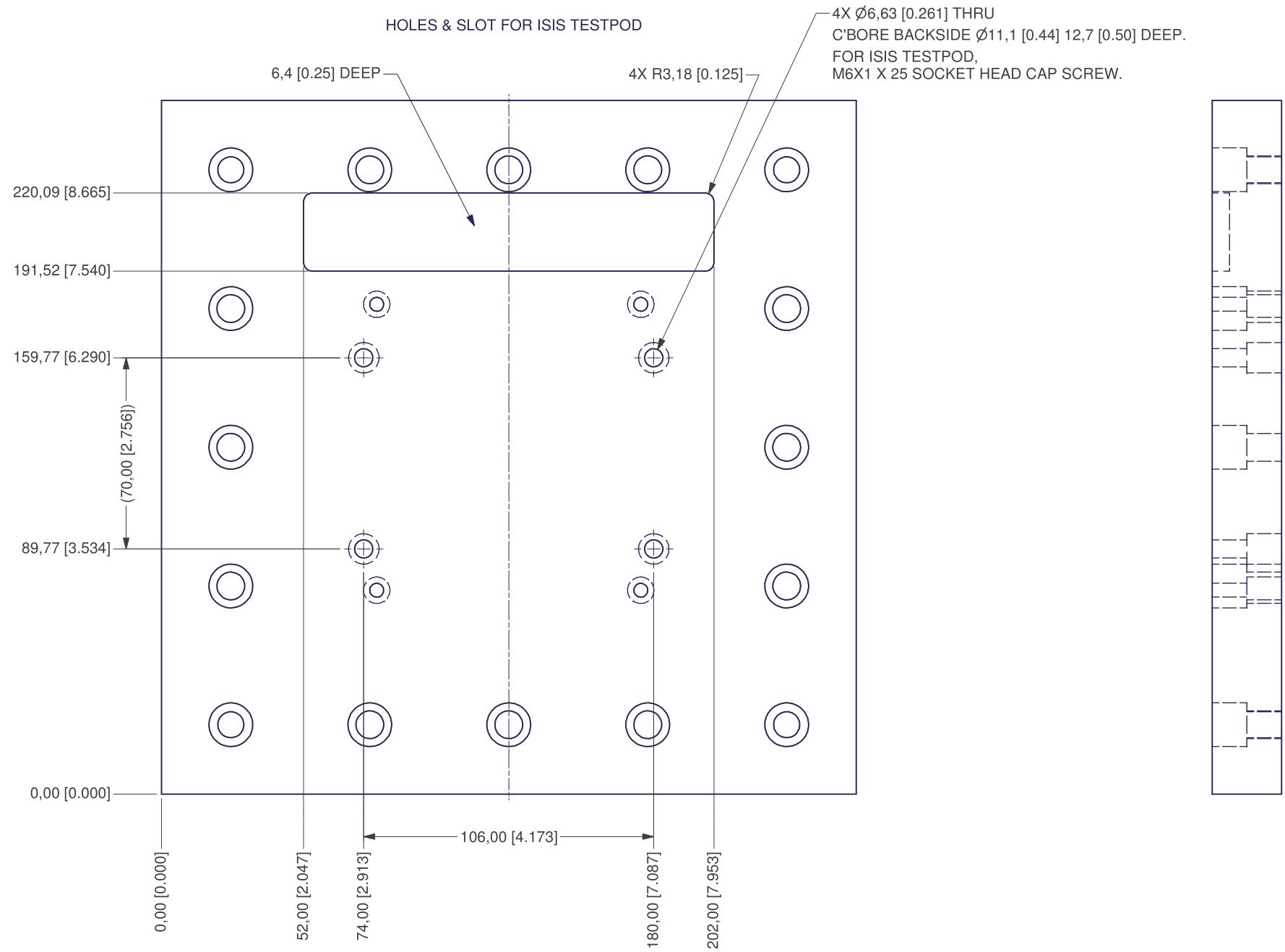
VIBRATION ADAPTER

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2015-12-2 | FOX-ME-140 | 0.1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | SHEET 1 OF 3 | |



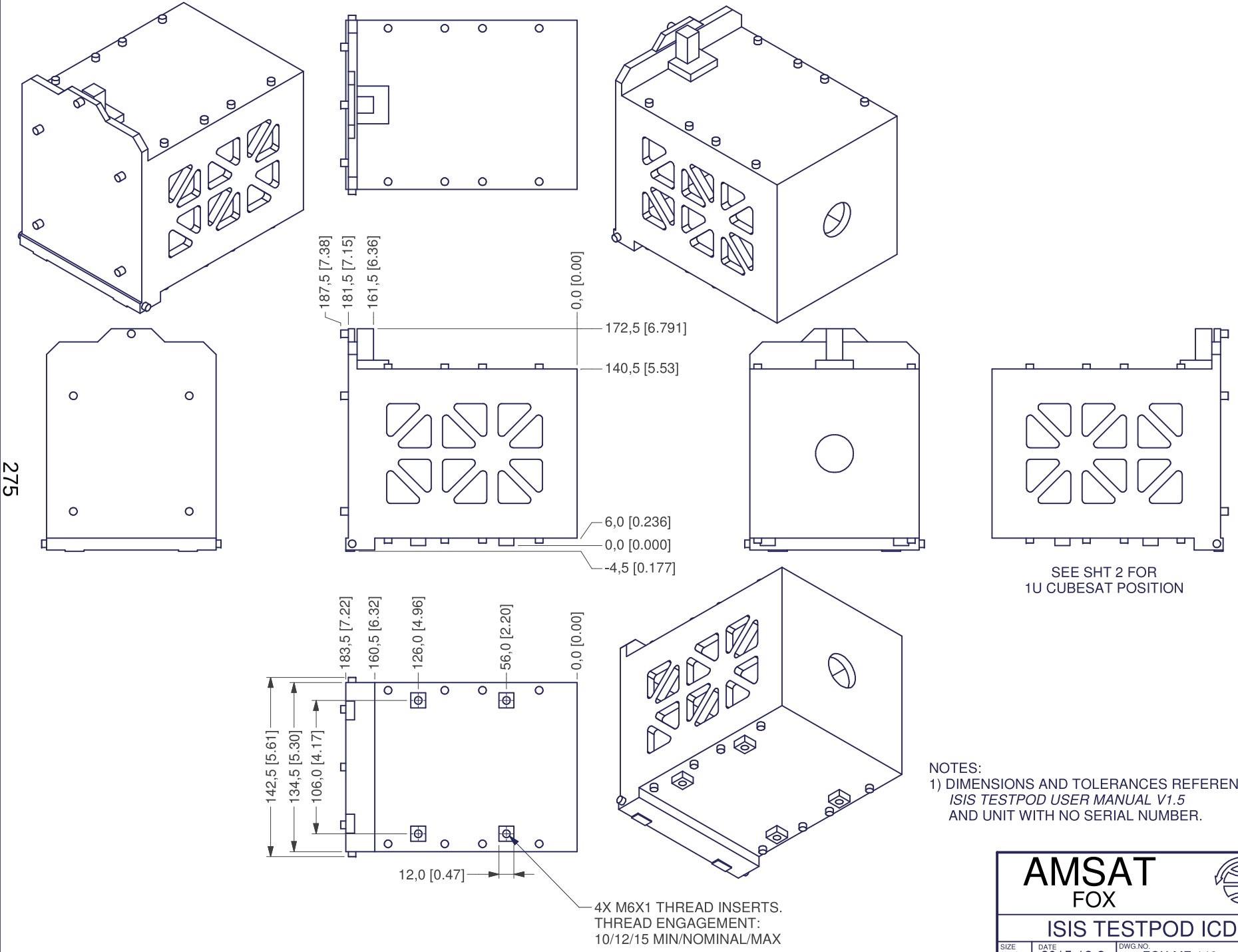

273

| VIBRATION ADAPTER | | | |
|---|---|---|---|
| SIZE A | DATE 2015-12-2 | DWG.NO. FOX-ME-140 | REV. 0.1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 3 |




274

| VIBRATION ADAPTER | | | |
|---|---|---|---|
| SIZE A | DATE 2015-12-2 | DWG.NO. FOX-ME-140 | REV. 0.1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 3 |




275

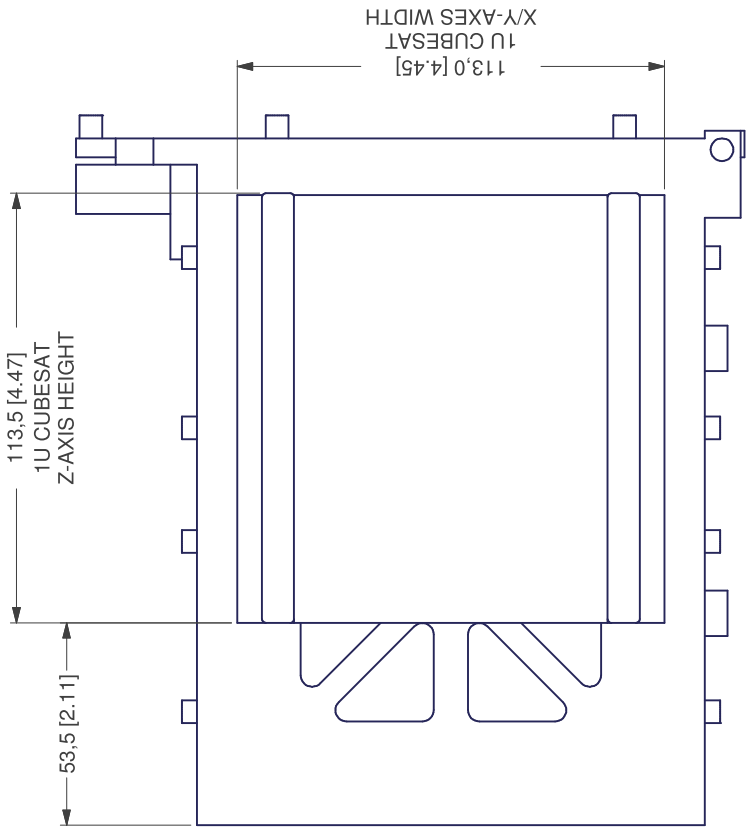
NOTES:
1) DIMENSIONS AND TOLERANCES REFERENCES:
*ISIS TESTPOD USER MANUAL V1.5*
AND UNIT WITH NO SERIAL NUMBER.

AMSAT
FOX

# ISIS TESTPOD ICD

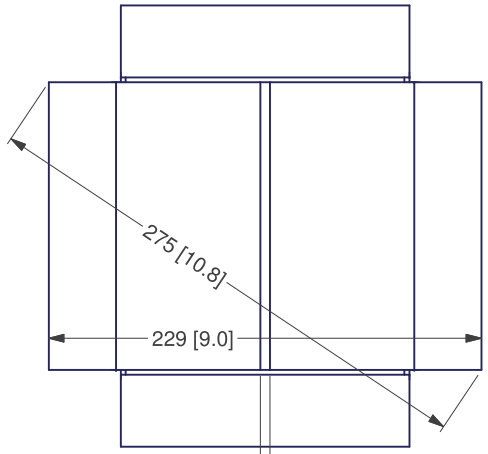
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2015-12-2 | FOX-ME-142 | 0.1 |

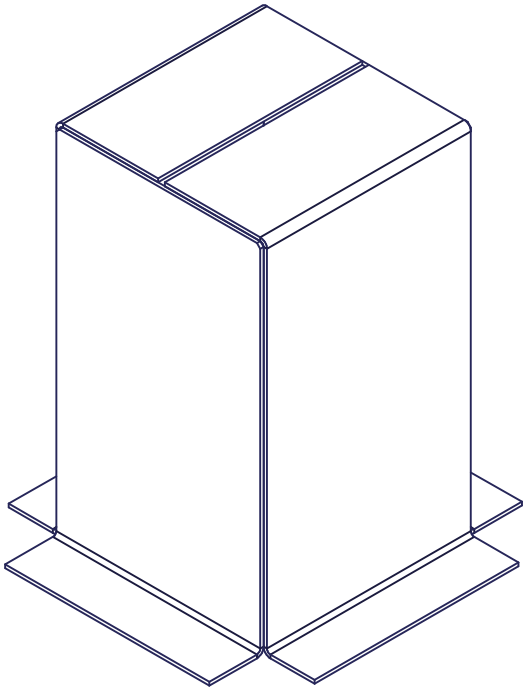
| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 4 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 2 |

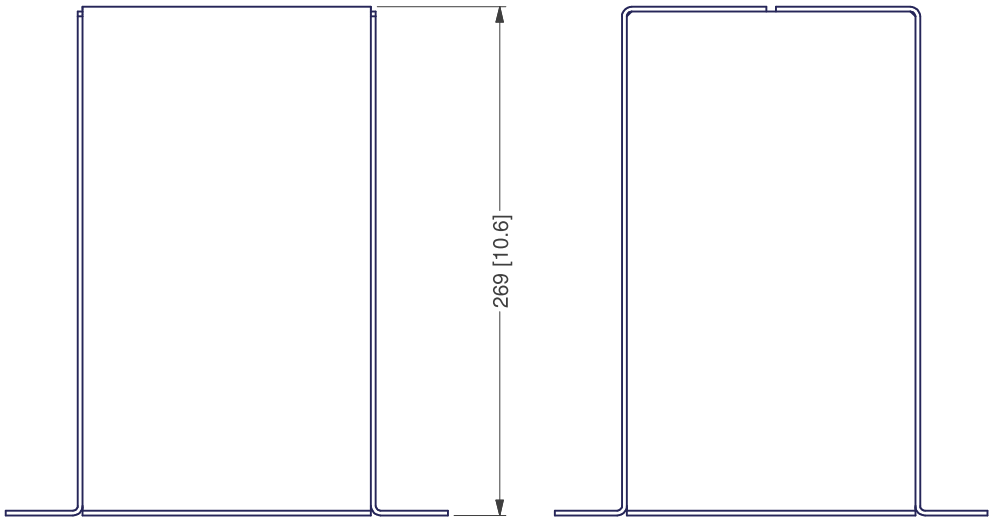



POSITION OF 1U CUBESAT ENVELOPE
WITHIN ISIS TESTPOD.

# -01 HAT ASSEMBLY AT THERMAL VACUUM BAKEOUT



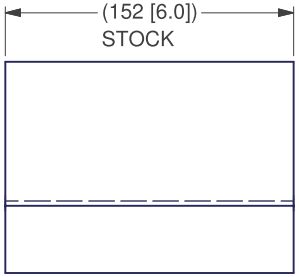


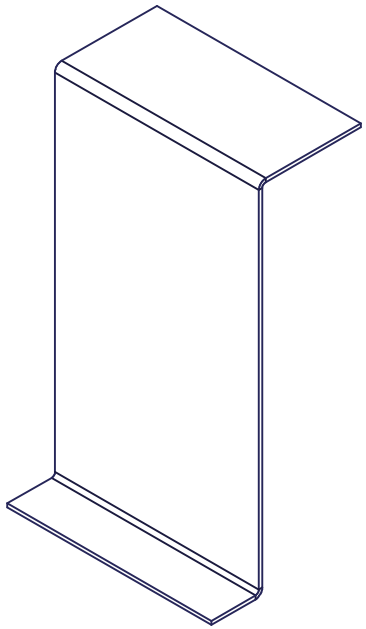

277

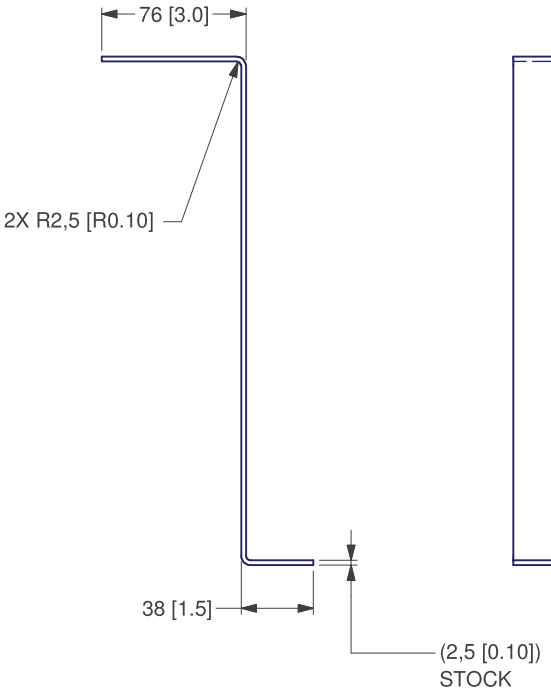
NOTES:
1) HAT PIECES ARE CLAMPED TO BASE PLATE PROVIDED BY TEST FACILITY.
RADIATIVELY HEATED OR HEATER STRIPS ATTACHED TO EXTERIOR.

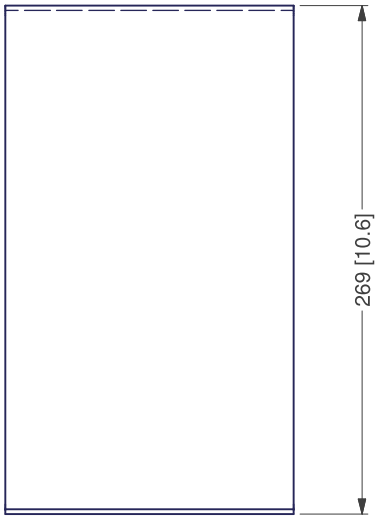
| | | | |
|---|---|---|---|
| AMSAT FOX | | | |
| HAT, 2U BAKEOUT | | | |
| SIZE A | DATE 2015-01-23 | DWG.NO. FOX-ME-143 | REV. 0.1 |
| SCALE 1 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | | SHEET 1 OF 3 |

# -03 WALL WITH PARTIAL LID





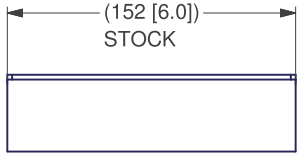



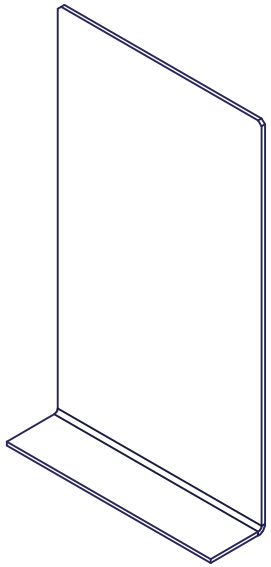

NOTES:
2) MATERIAL: ALUMINUM 5052 H32
MAKE FROM MCMASTER-CARR 88895K28

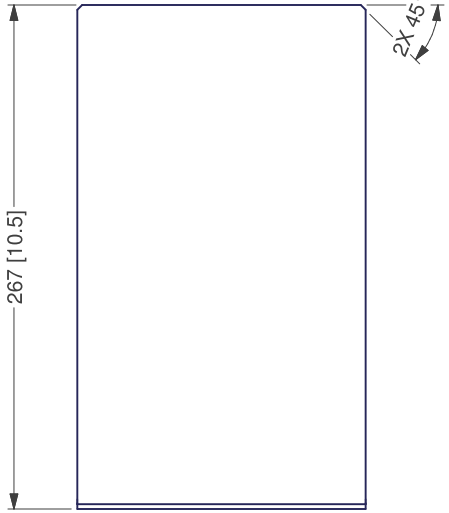
278

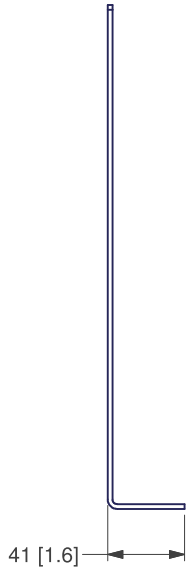
| HAT, 2I BAKEOUT | | | |
|---|---|---|---|
| SIZE A | DATE 2015-01-23 | DWG.NO. FOX-ME-143 | REV. 0.1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 3 |

# -05 WALL



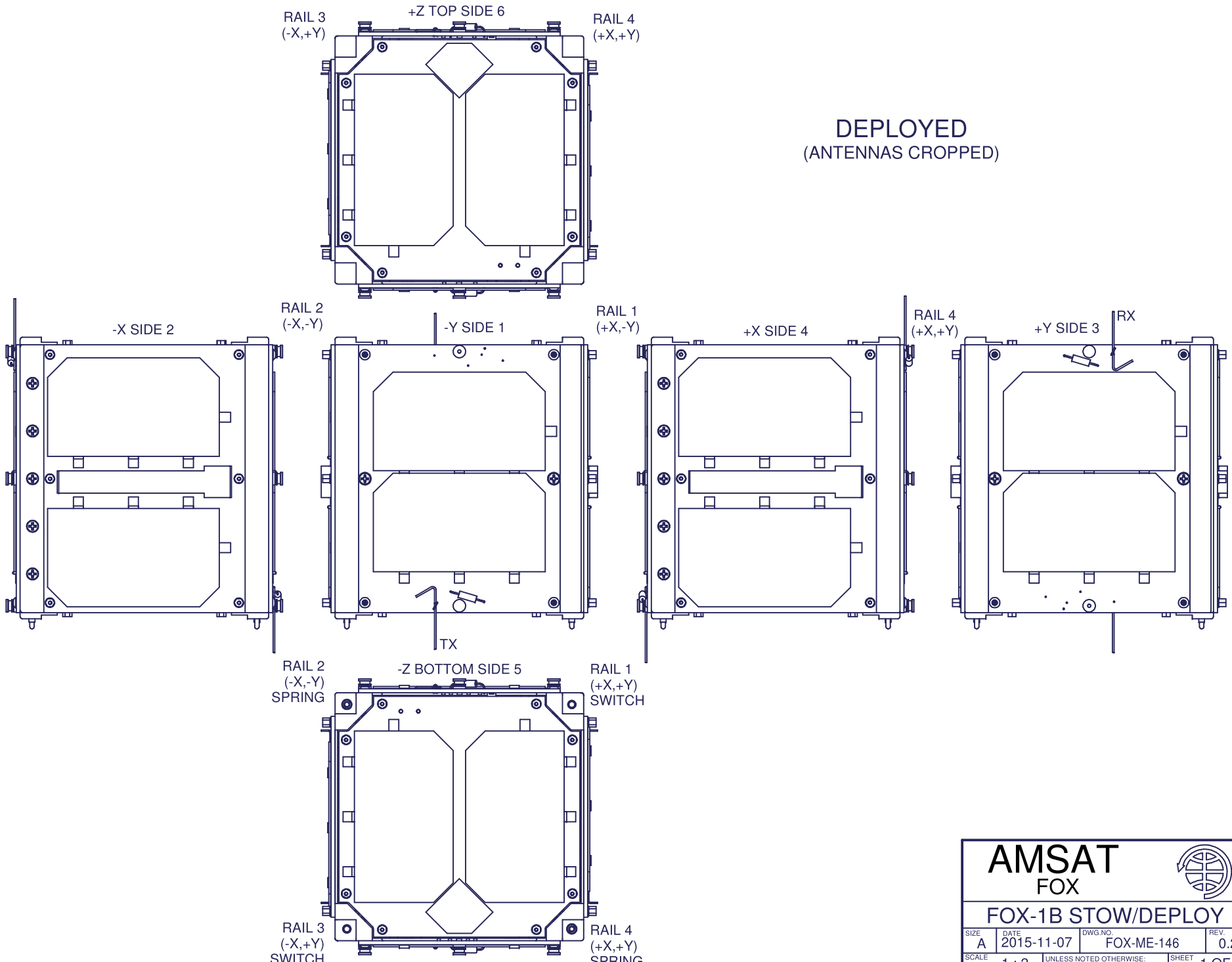



279

NOTES:
3) MATERIAL: ALUMINUM 5052 H32
MAKE FROM MCMASTER-CARR 88895K28

| HAT, 2U BAKEOUT | | | |
|---|---|---|---|
| SIZE A | DATE 2015-01-23 | DWG.NO. FOX-ME-143 | REV. 0.1 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 3 |

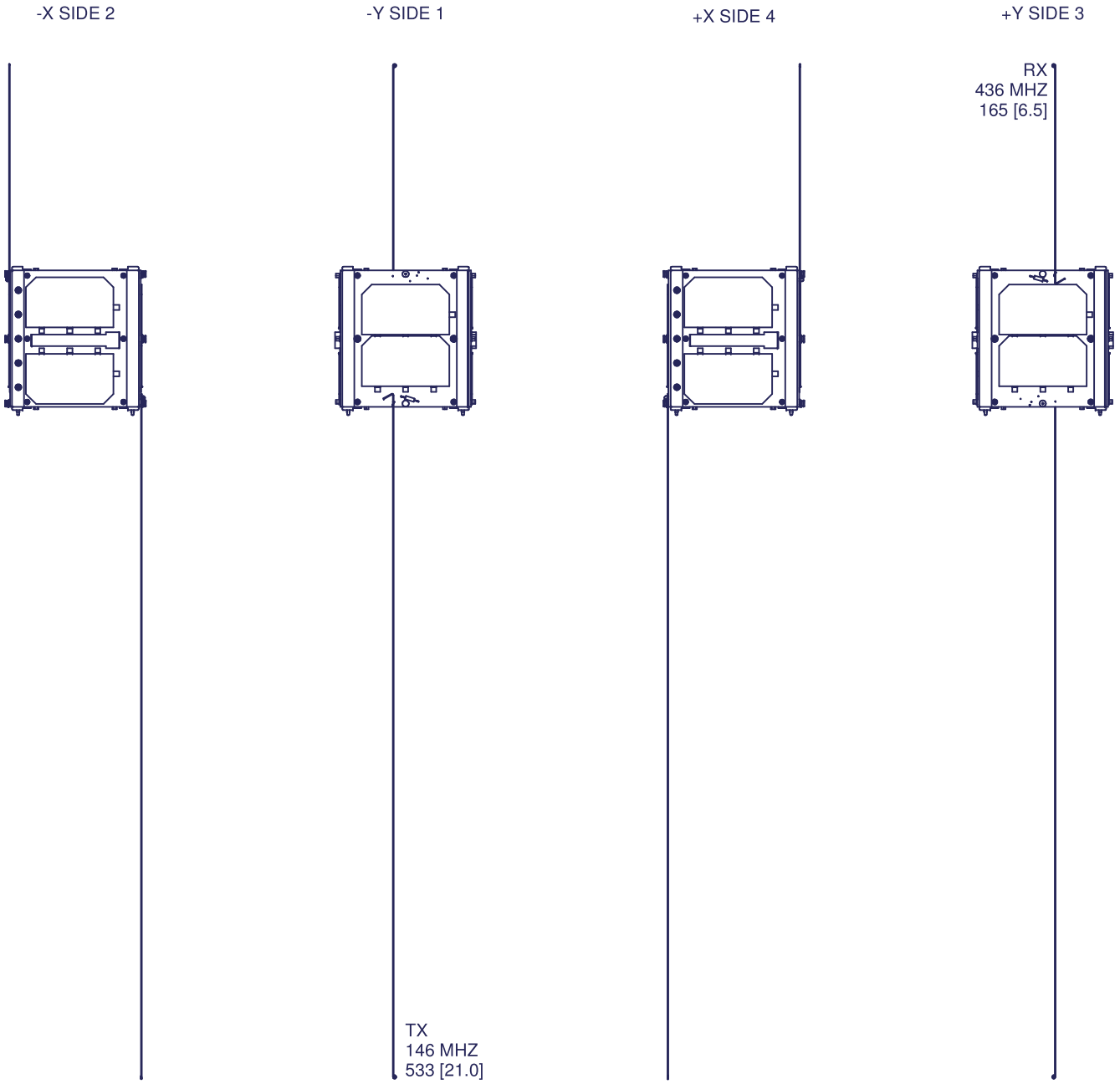



AMSAT
FOX

FOX-1B STOW/DEPLOY

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2015-11-07 | FOX-ME-146 | 0.2 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 6 |

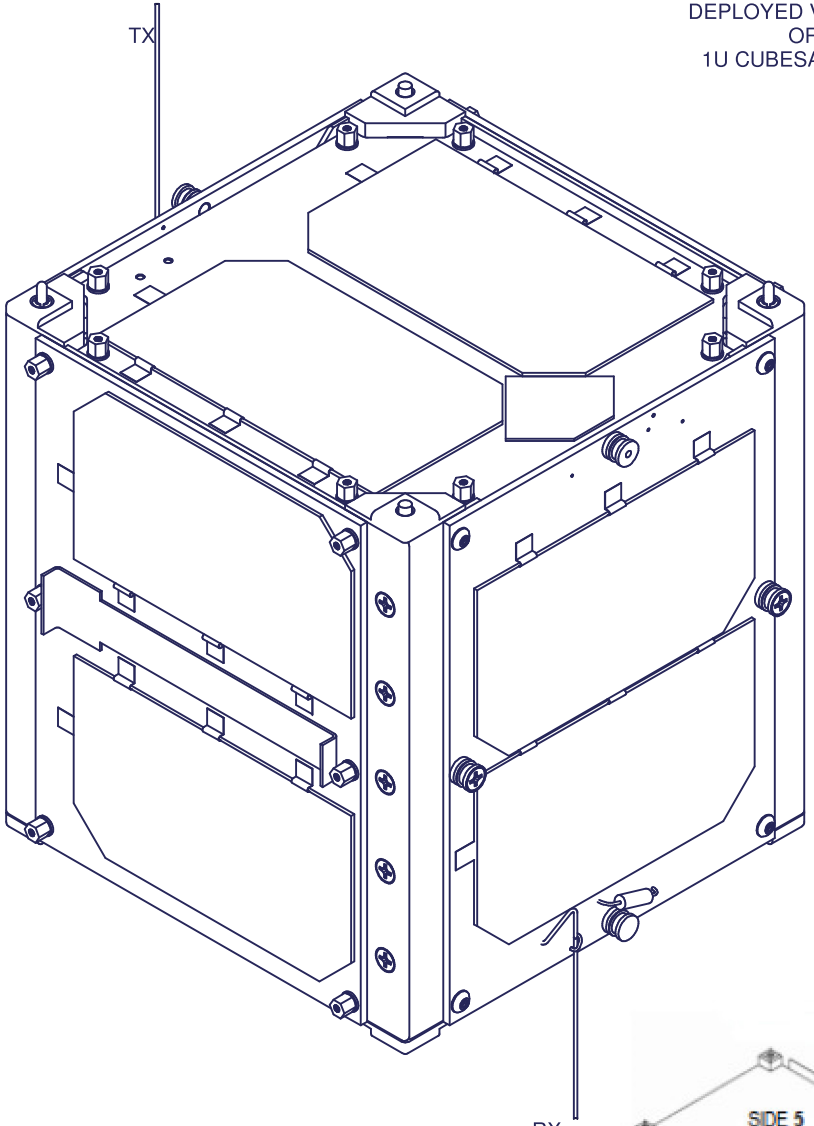


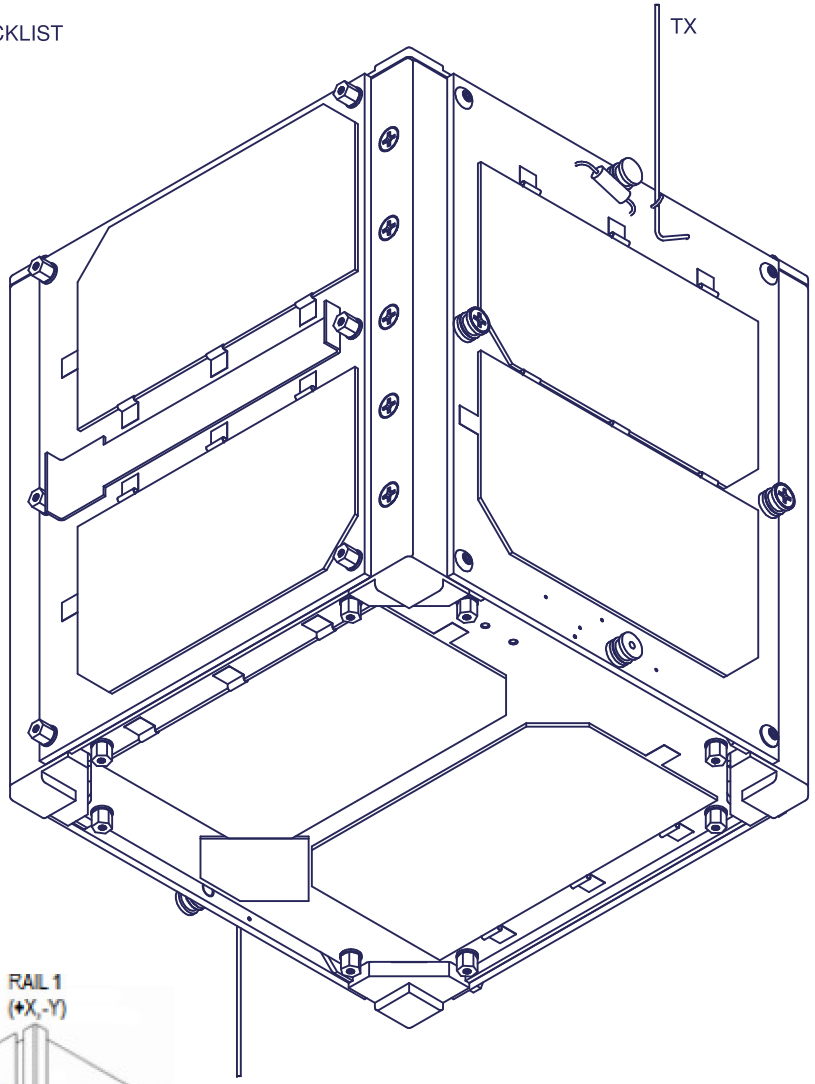

# DEPLOYED
(ENTIRE ANTENNAS)

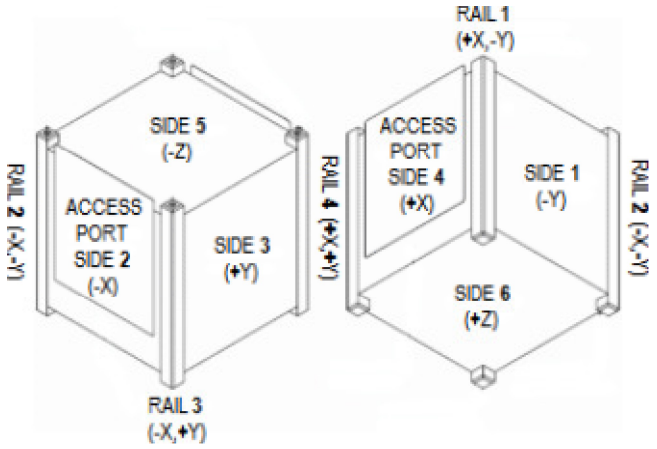
281

| FOX-1B STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-146 | REV. 0.2 |
| SCALE 1 : 5 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 6 |

DEPLOYED VIEWS (ANTENNAS CROPPED)
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST

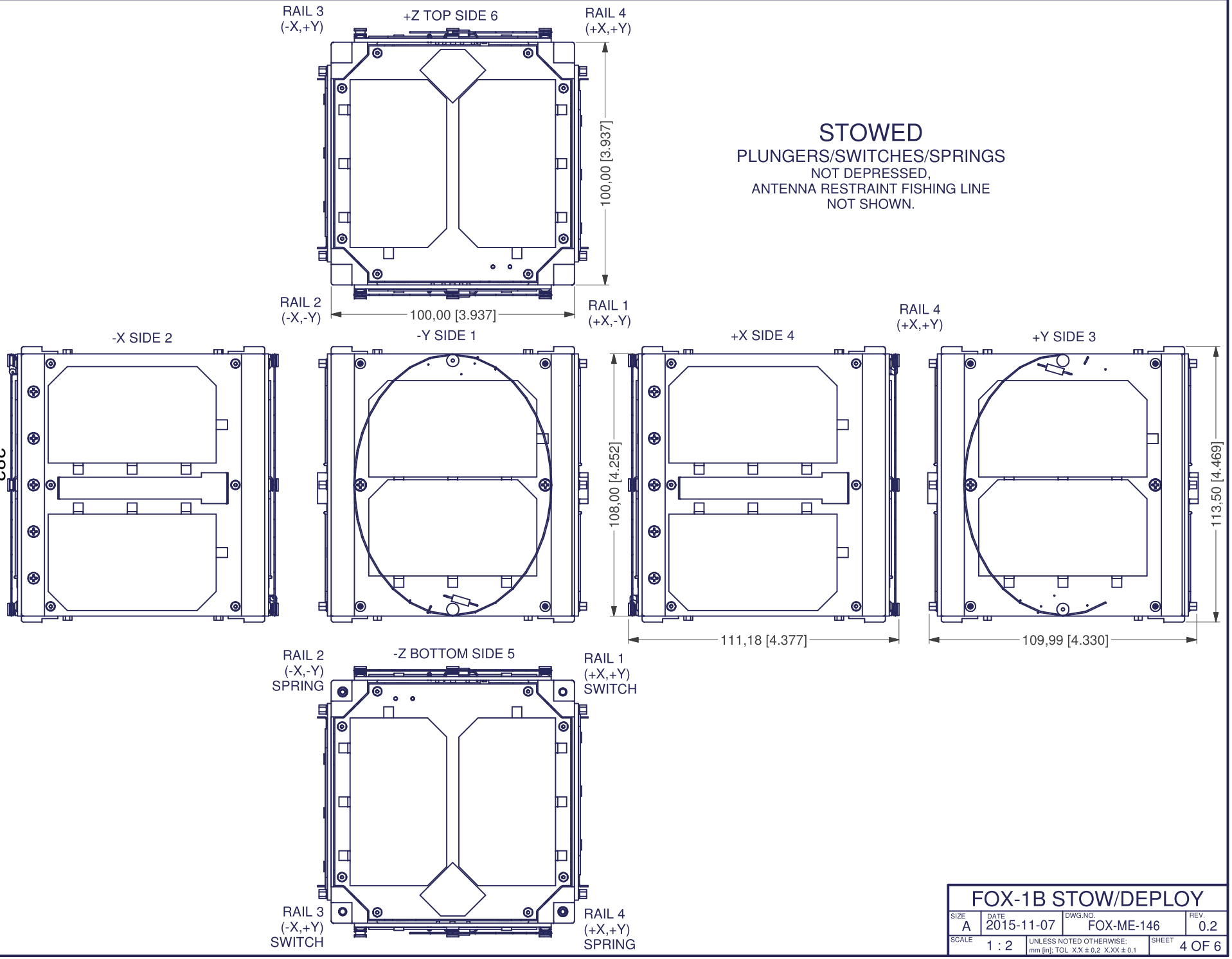



282

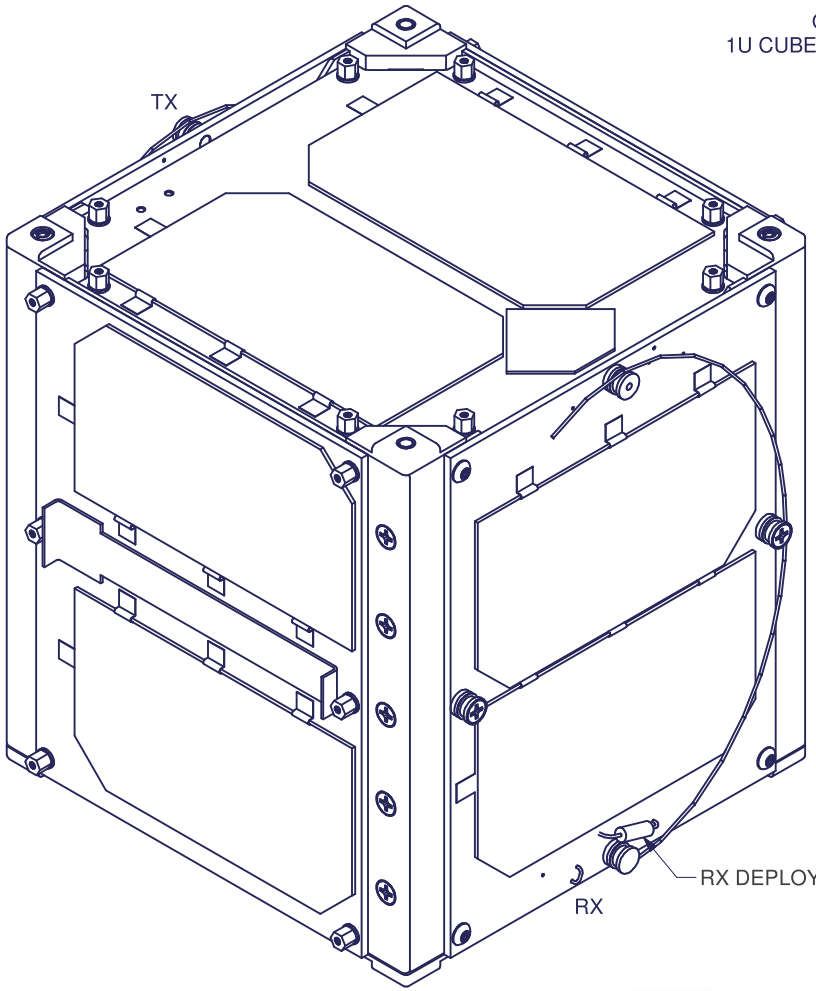
| FOX-1B STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-146 | REV. 0.2 |
| SCALE 3 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 6 |

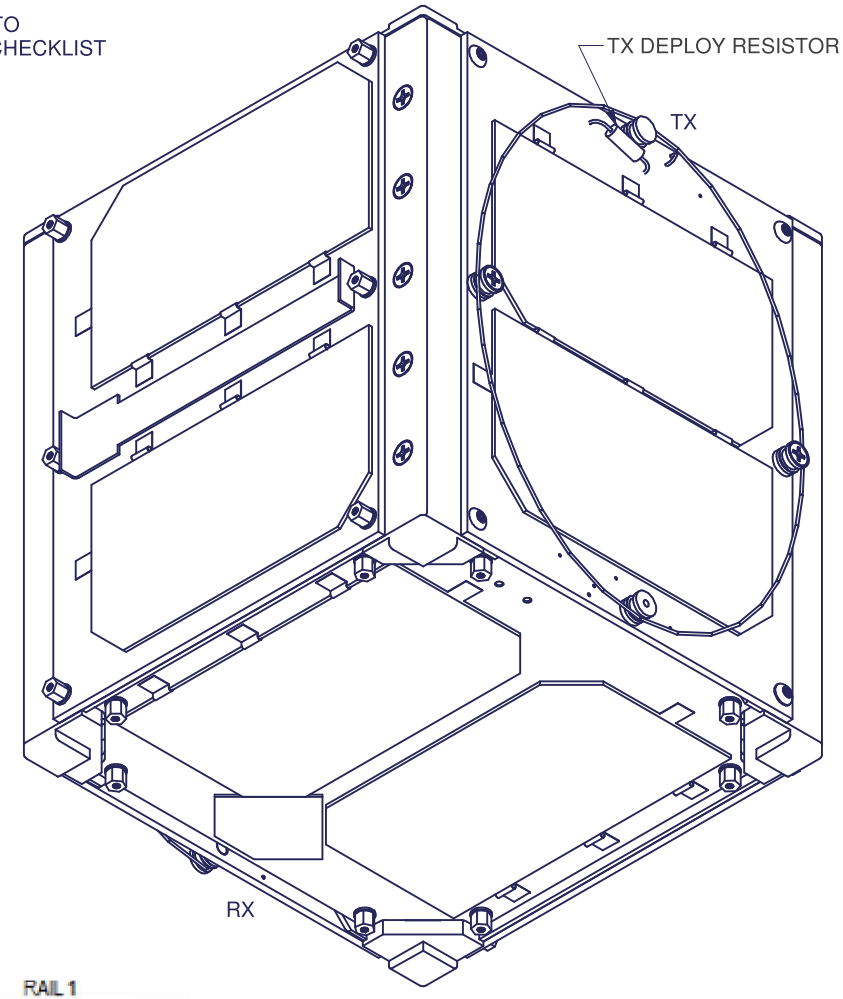


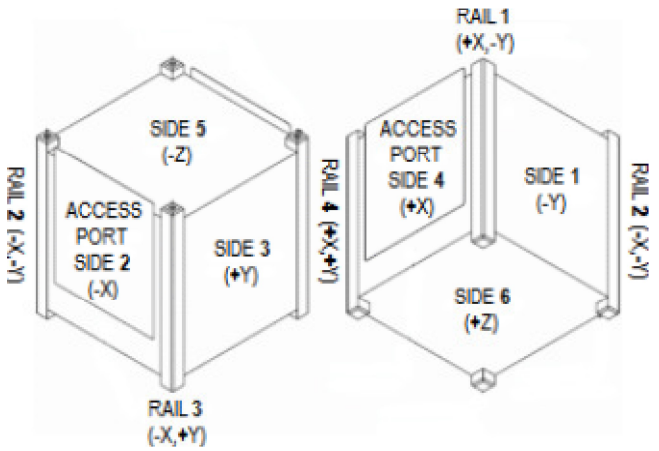

283

| FOX-1B STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-146 | REV. 0.2 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | SHEET 4 OF 6 | |

STOWED VIEWS
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST

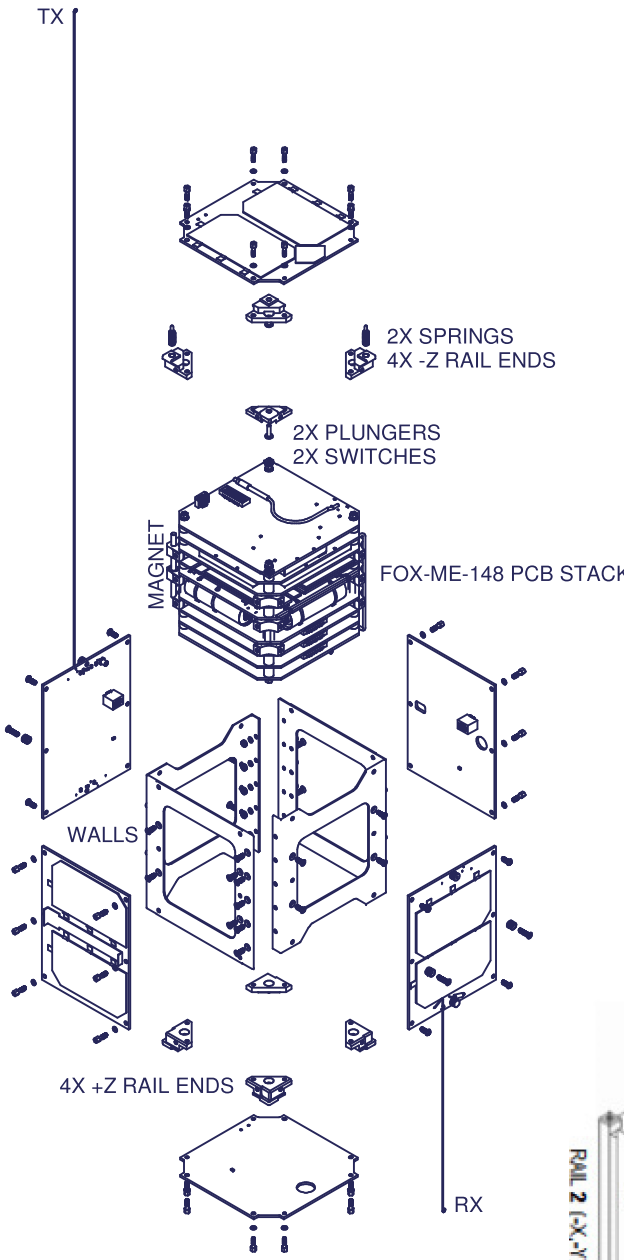


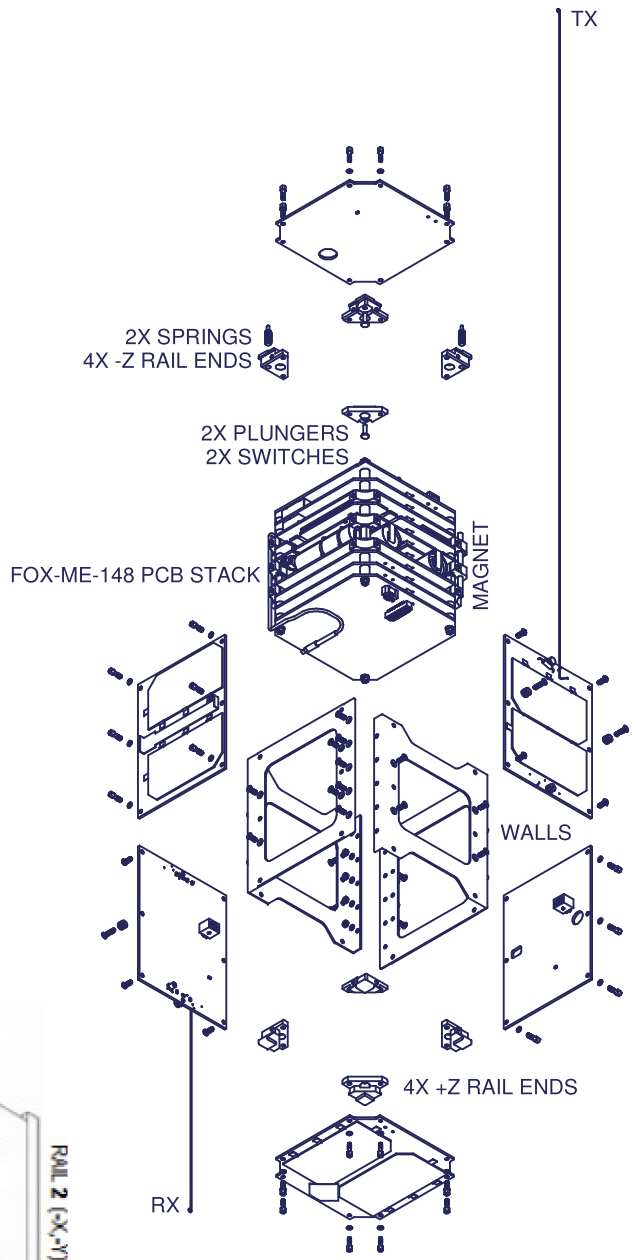



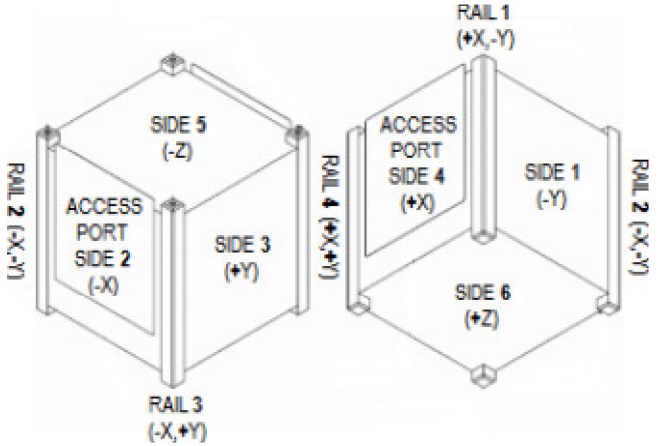




| FOX-1B STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-146 | REV. 0.2 |
| SCALE 3 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | SHEET 5 OF 6 | |

EXPLODED VIEWS
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST

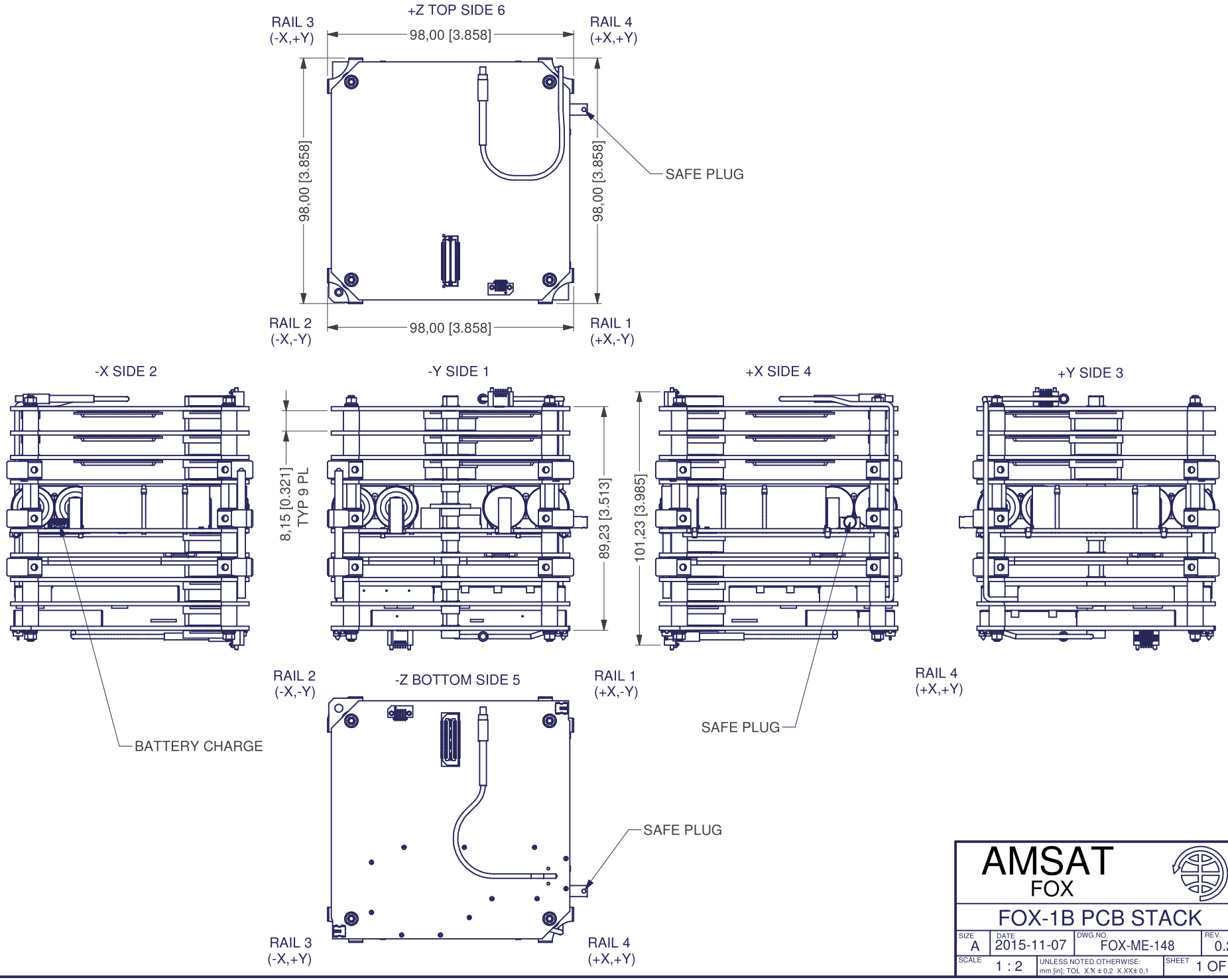



285

| FOX-1B STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-146 | REV. 0.2 |
| SCALE 1 : 5 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 6 OF 6 |




286

AMSAT
FOX

FOX-1B PCB STACK

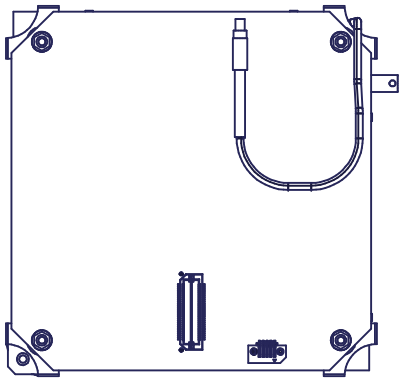
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2015-11-07 | FOX-ME-148 | 0.2 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 8 |

RAIL 3
(-X,+Y)

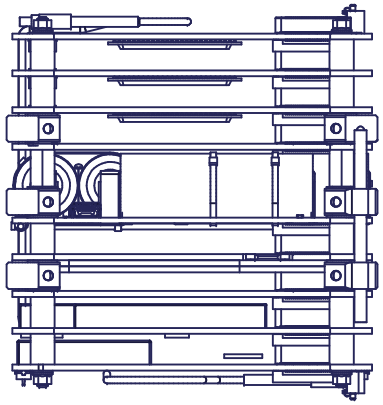
+Z TOP SIDE 6

RAIL 4
(+X,+Y)

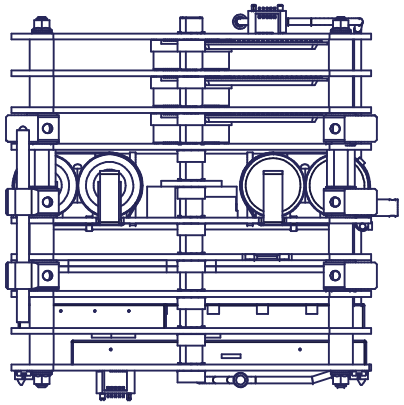


RAIL 2
(-X,-Y)

-X SIDE 2

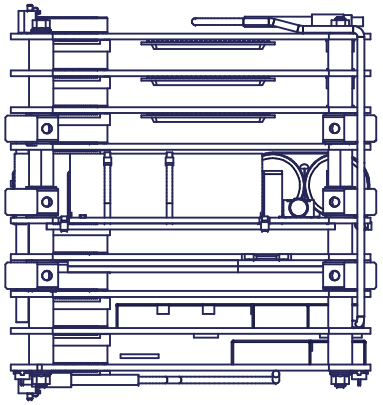


-Y SIDE 1

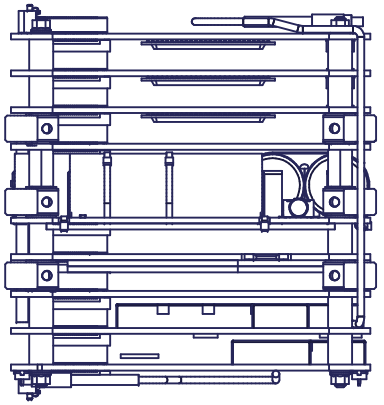


RAIL 1
(+X,-Y)

+X SIDE 4



+Y SIDE 3

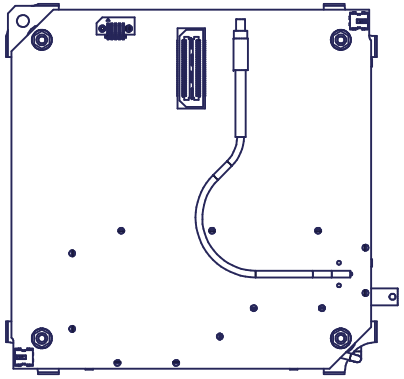


RAIL 2
(-X,-Y)

-Z BOTTOM SIDE 5

RAIL 1
(+X,-Y)

RAIL 4
(+X,+Y)

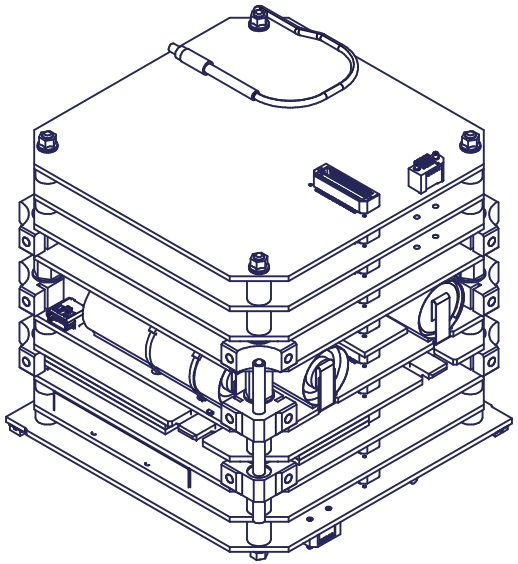


RAIL 3
(-X,+Y)

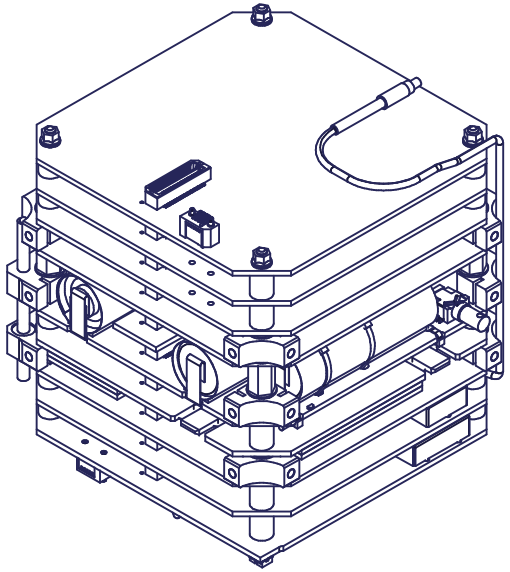
RAIL 4
(+X,+Y)

287

| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 8 |

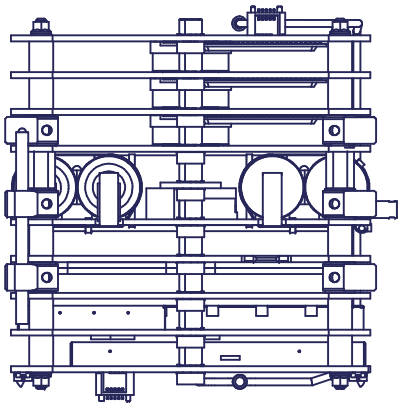


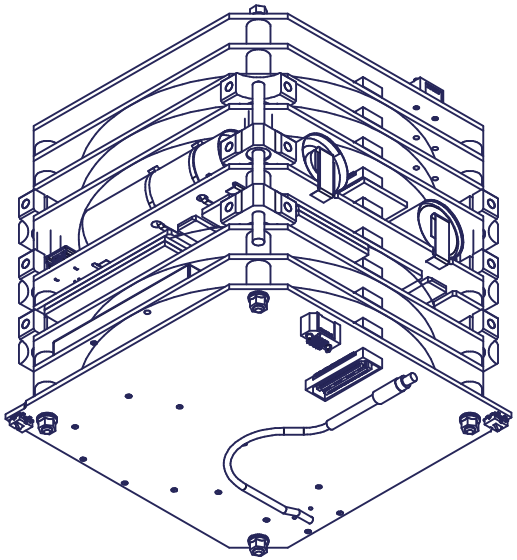
-X -Y +Z



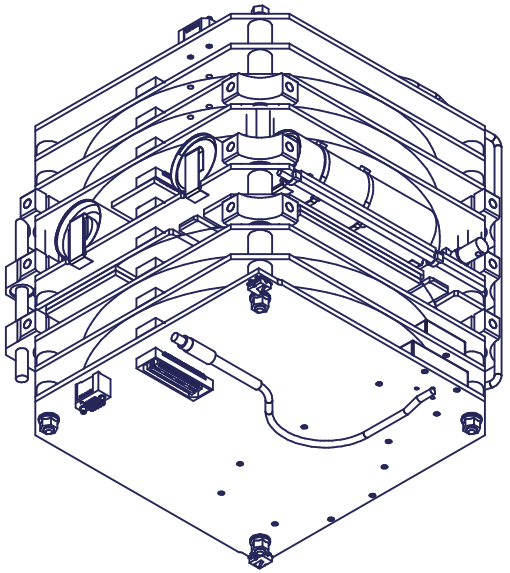
+X -Y +Z

-Y SIDE 1





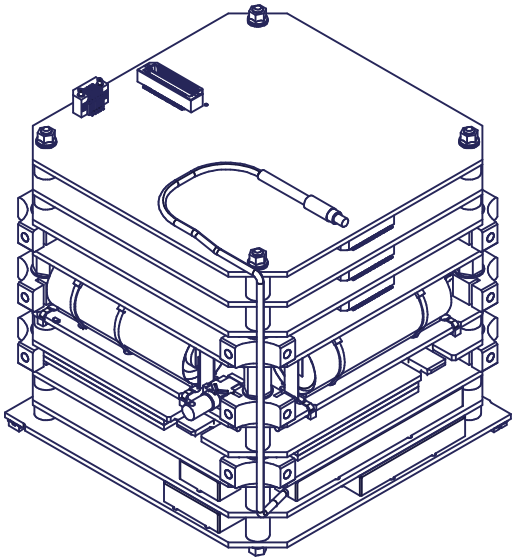
-X -Y -Z



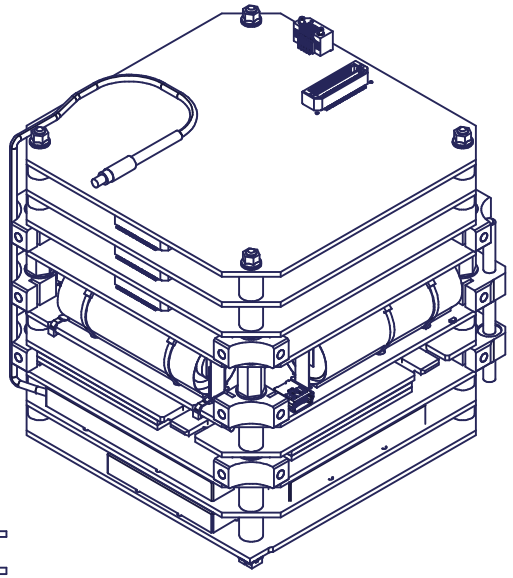
+X -Y -Z

288

| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 3 OF 8 |

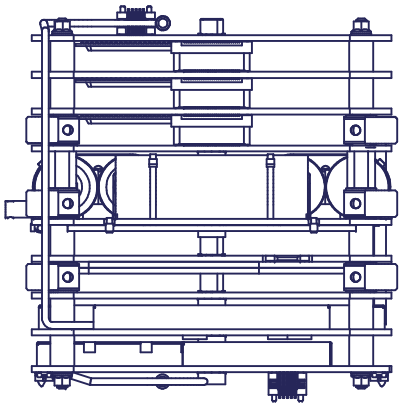


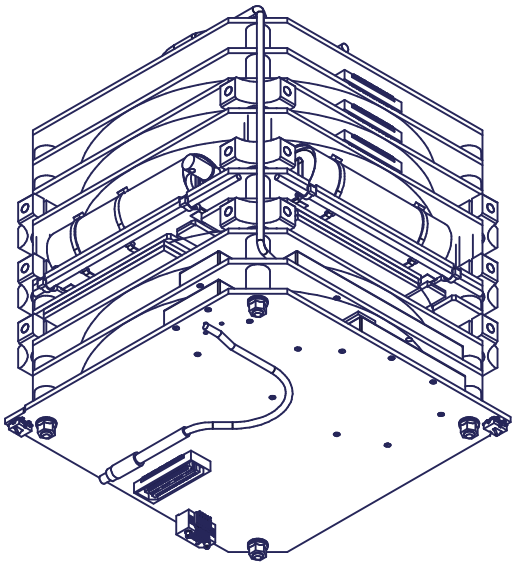
+X +Y +Z



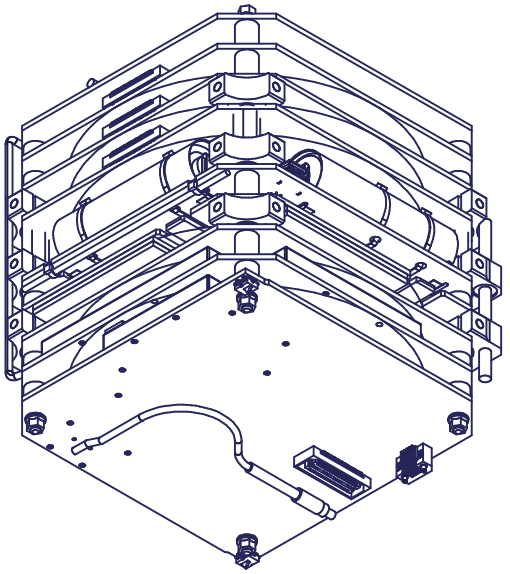
-X +Y +Z

+Y SIDE 3





+X +Y -Z

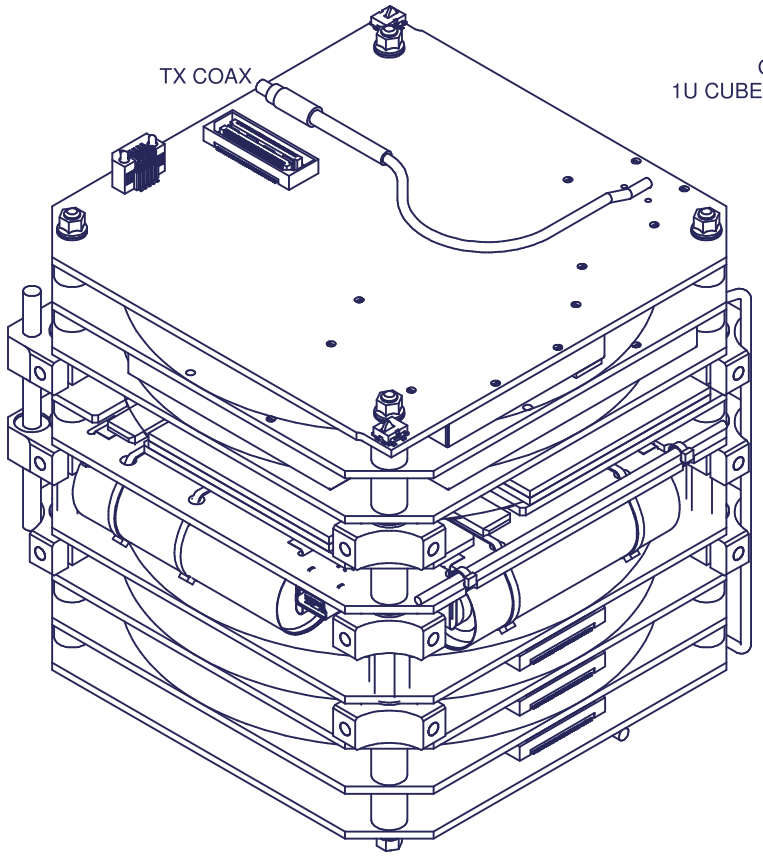


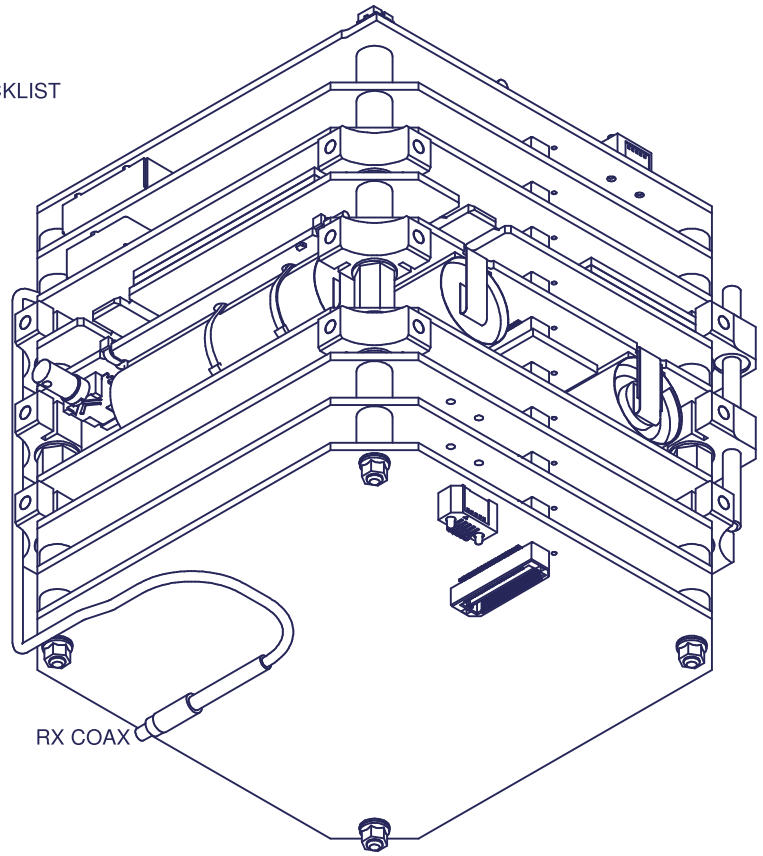
-X +Y -Z

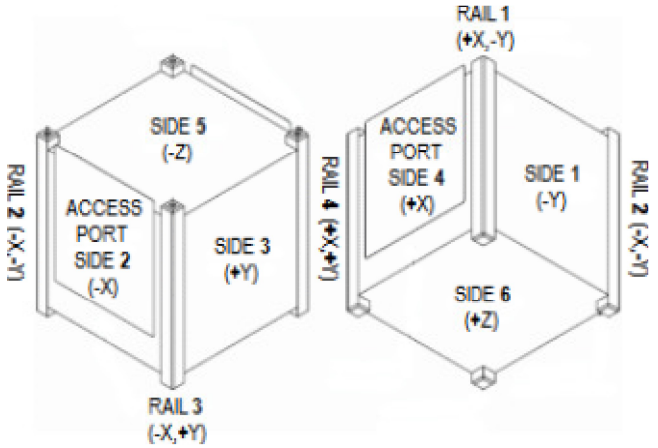
289

| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 4 OF 8 |

VIEWS
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST



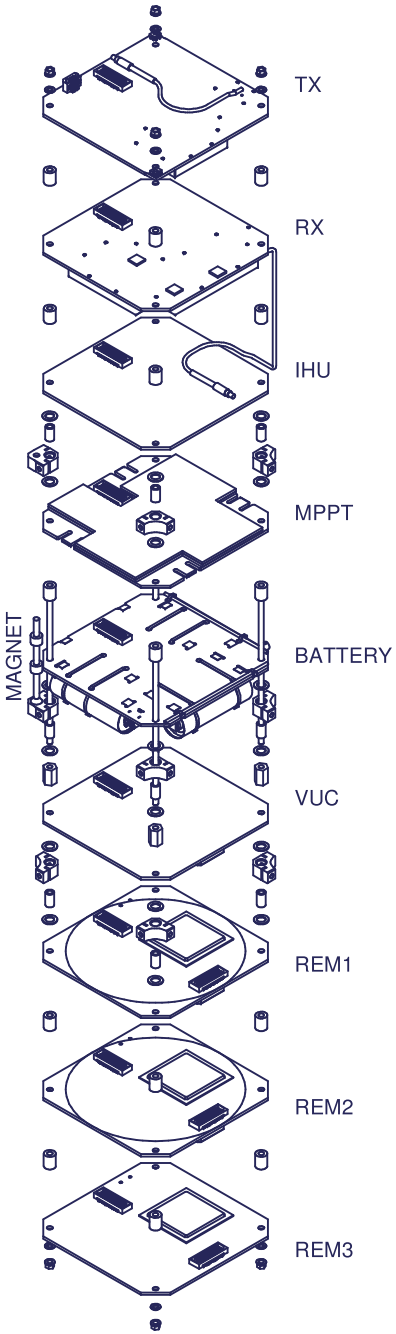



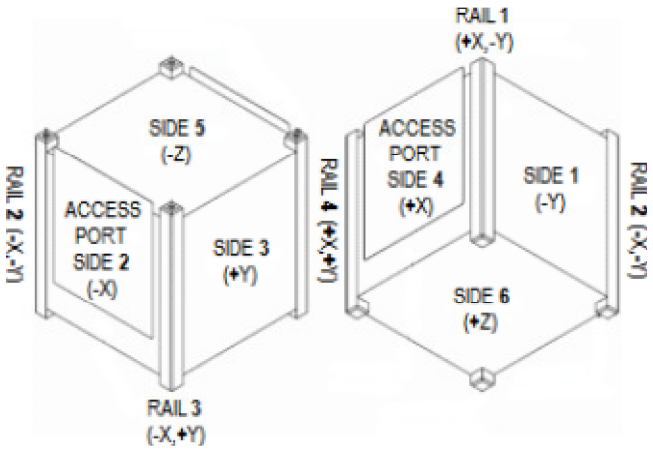



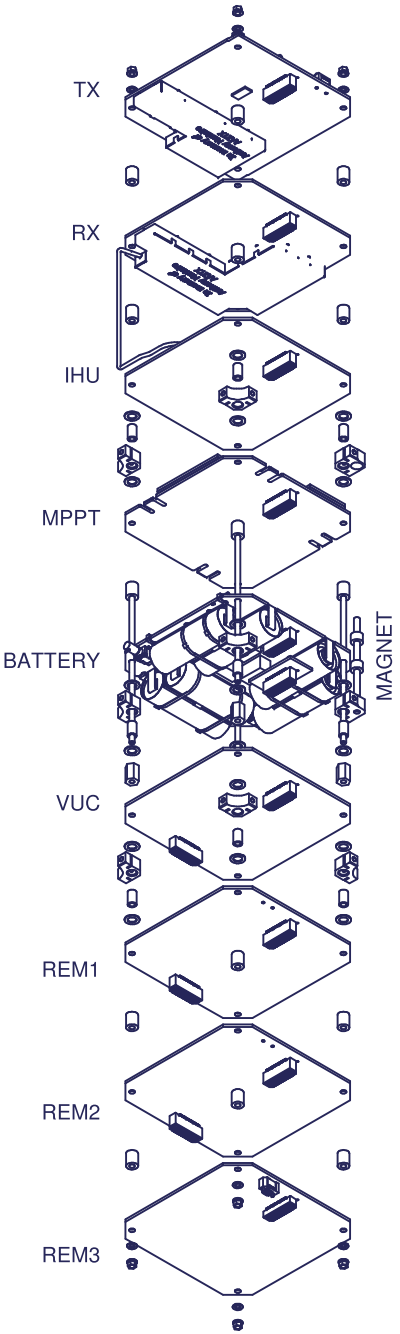

| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 2 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 5 OF 8 |

290

EXPLODED VIEWS
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST



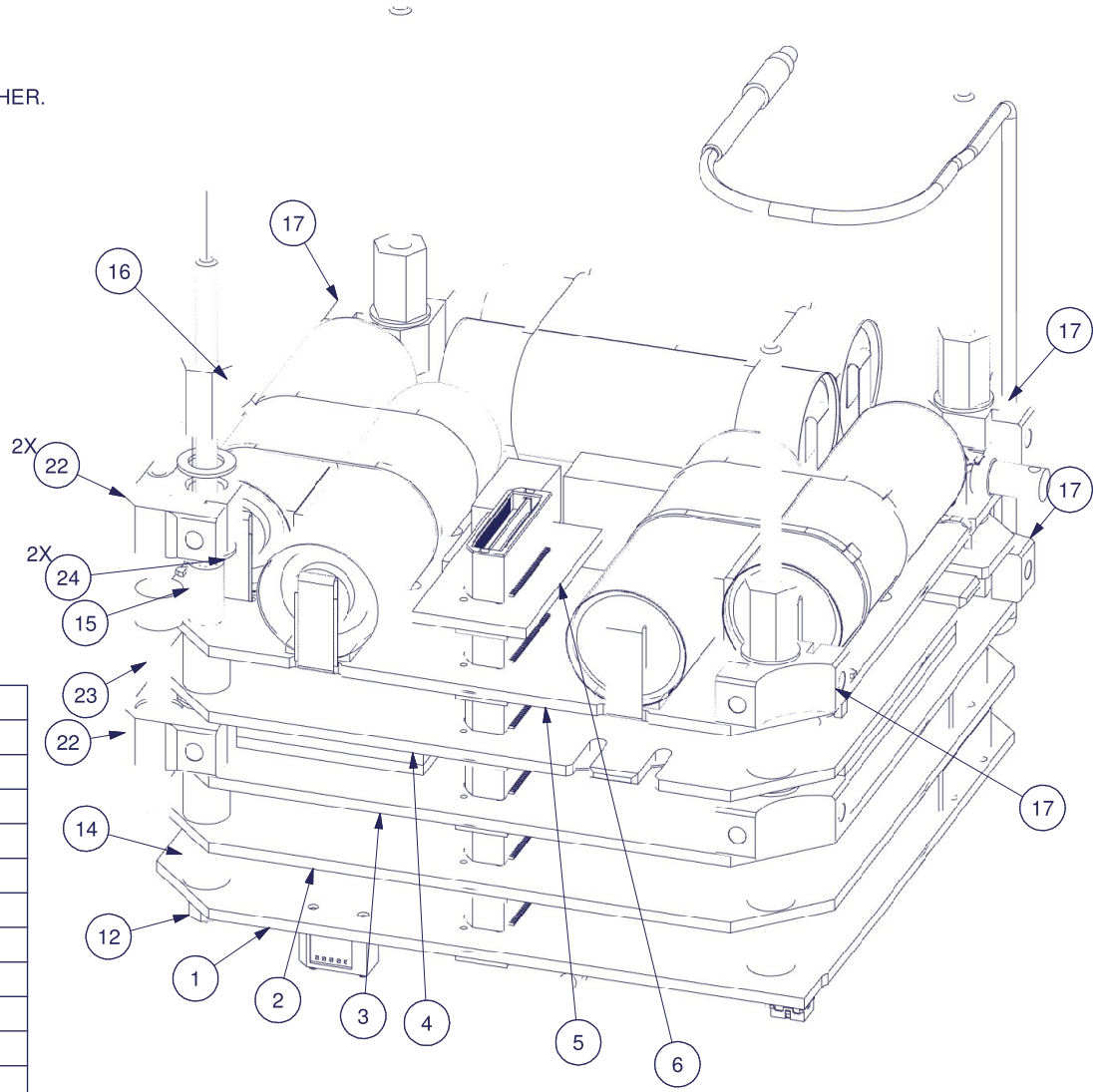




| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 6 OF 8 |

## ASSEMBLY OF AVIONICS PORTION OF PCB STACK

1) LOCATE ITEM 12 ON ITEM 11 TO PROTRUDE 4,5 [0.18] FROM PCB.
2) VERTICAL SPACING BETWEEN DELRIN MOUNTS MAY BE ADJUSTED
   BY USING 3X OR 1X 0.010 SHIM WASHER(S) FOR EACH 0.020 SHIM WASHER.
3) TORQUE ITEM 16 TO TBD IN-LB.
4) ITEM 6 IS CAPTURED BUT HAS NO HARDWARE.

THIS AREA SHOWN EXPLODED
FOR CLARITY OF SPACER INSIDE DELRIN
AND SHIM WASHERS ABOVE AND BELOW.



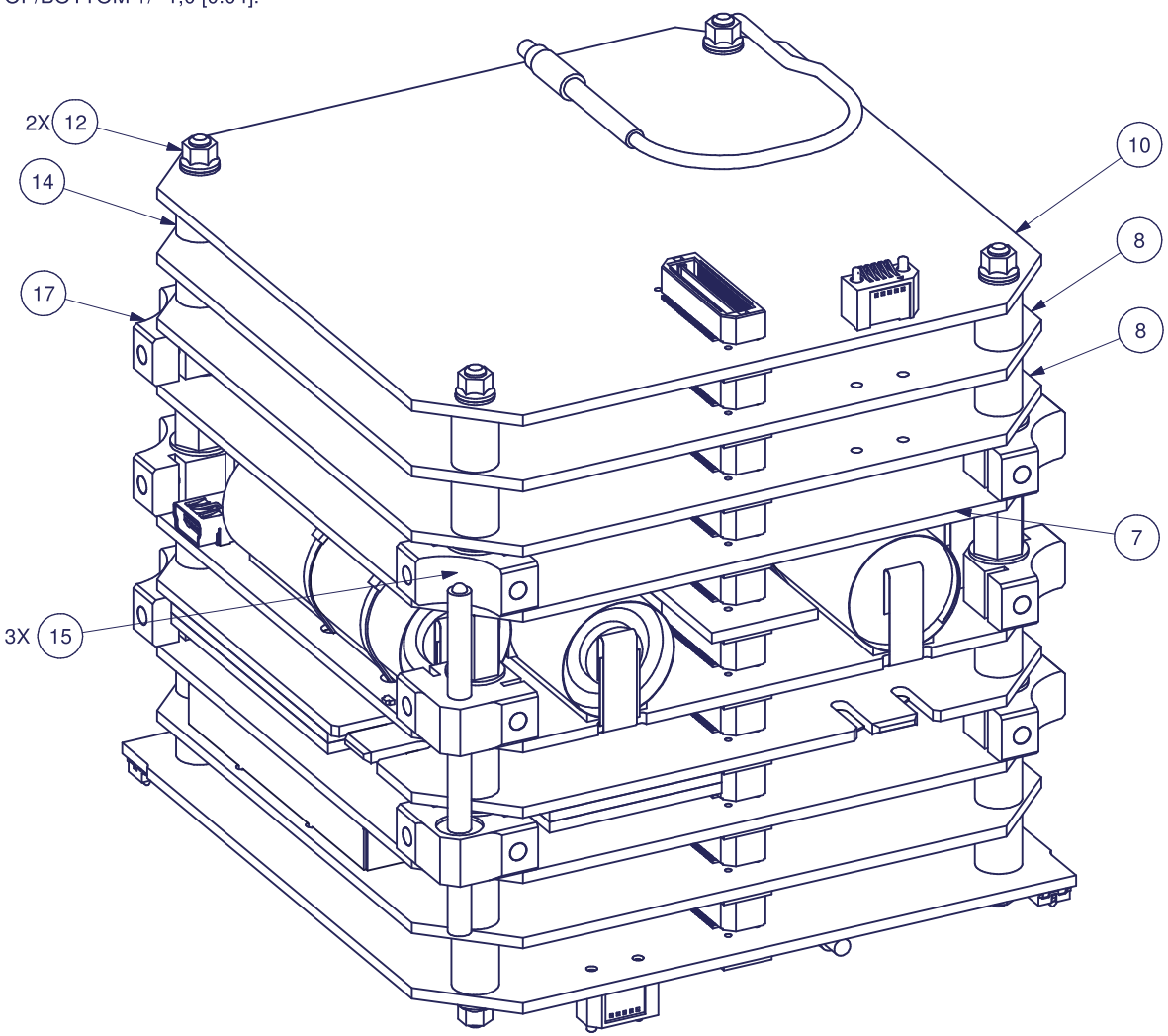

| ITEM | QTY | PART NUMBER | DESCRIPTION |
|---|---|---|---|
| 17 | 10 | FOX-ME-118-01 | PCB DELRIN MOUNT |
| 16 | 4 | FOX-ME-117-05 | AL THREADED SPACER |
| 15 | 12 | FOX-ME-117-03 | CU SPACER FOR DELRIN |
| 14 | 20 | FOX-ME-117-01 | AL SPACER |
| 13 | 8 | NAS620-C4L | WASHER, #4 THIN |
| 12 | 8 | MS21042-04 | LOCKING NUT, #4-40 |
| 11 | 4 | FOX-ME-123 | THREADED ROD |
| 10 | 1 | FOX-ME-149-10 | EXPERIMENT #4 PCB |
| 8 | 2 | FOX-ME-149-08 | EXPERIMENTS #2 & #3 PCB |
| 7 | 1 | FOX-ME-149-07 | EXPERIMENT #1 PCB |
| 6 | 1 | FOX-ME-149-06 | BATTERY #2 PCB |
| 5 | 1 | FOX-ME-149-05 | BATTERY #1 |
| 4 | 1 | FOX-ME-149-04 | MPPT PCB |
| 3 | 1 | FOX-ME-149-03 | IHU PCB |
| 2 | 1 | FOX-ME-149-02 | RX PCB |
| 1 | 1 | FOX-ME-149-01 | TX PCB |
| 21 | 1 | FOX-ME-130 | SAFE/ARM RBF |
| 22 | 2 | FOX-ME-118-03 | PCB DELRIN MOUNT |
| 23 | 1 | FOX-ME-124 | MAGNET ASSEMBLY |
| 24 | 24 | MCMASTER 94773A708 | SHIM WASHER, 3/16" ID X 0.020" |

Parts List

| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-07 | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 7 OF 8 |

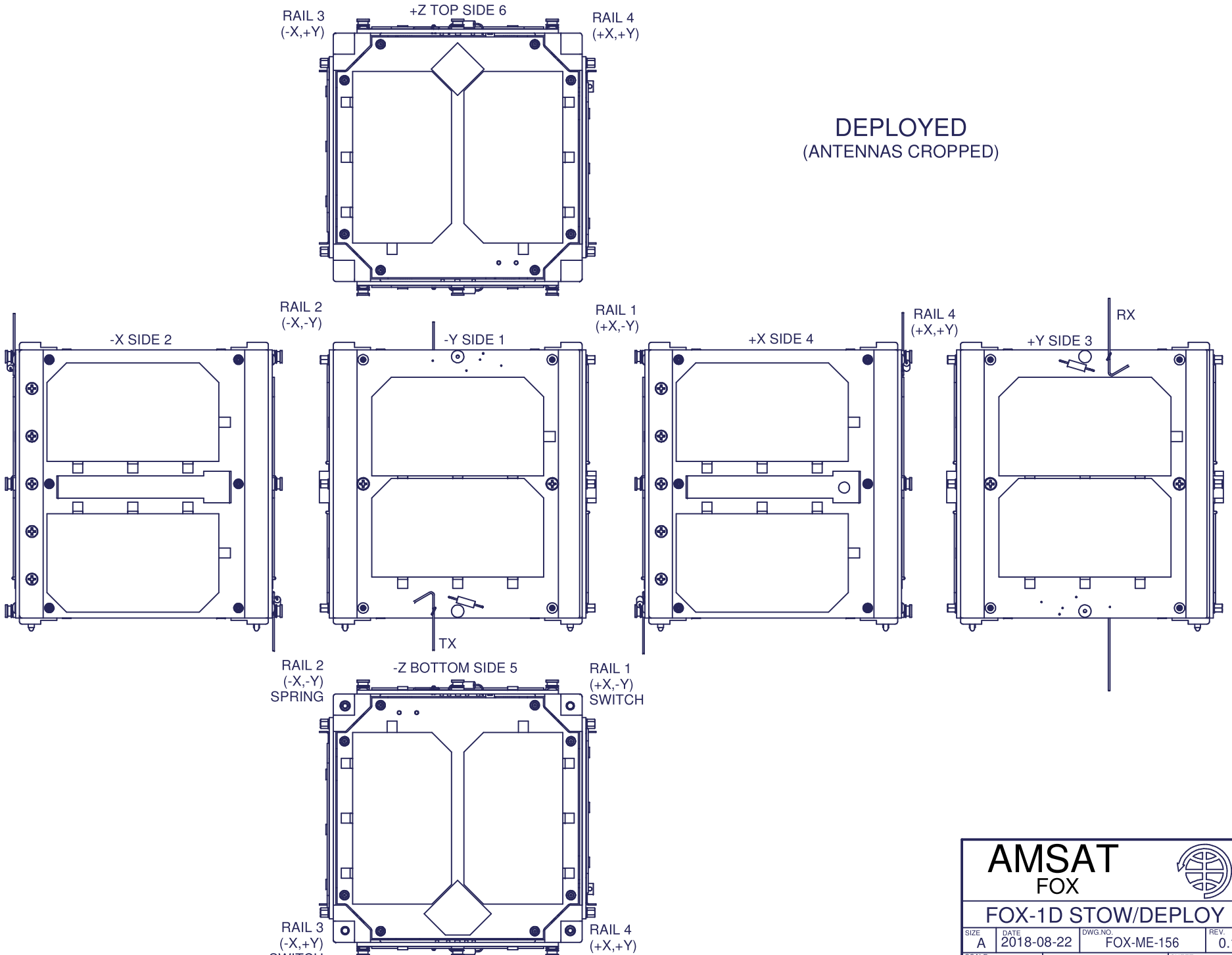
## ASSEMBLY OF AVIONICS PORTION OF PCB STACK

5) STACK PCBS ITEMS 7-10 AS SHOWN USING SPACERS ITEMS 14-16.
   VERTICAL SPACING BETWEEN DELRIN MOUNTS MAY BE ADJUSTED
   BY USING 3X OR 1X 0.010 SHIM WASHER(S) FOR EACH 0.020 SHIM WASHER.
6) TORQUE ITEM 16 TO TBD IN-LB.
7) VERIFY PROTRUSION OF THREADED ROD IS SAME TOP/BOTTOM +/- 1,0 [0.04].






| FOX-1B PCB STACK | | | |
|---|---|---|---|
| SIZE A | DATE 2015-11-0Y | DWG.NO. FOX-ME-148 | REV. 0.2 |
| SCALE 1 : 1 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 8 OF 8 |



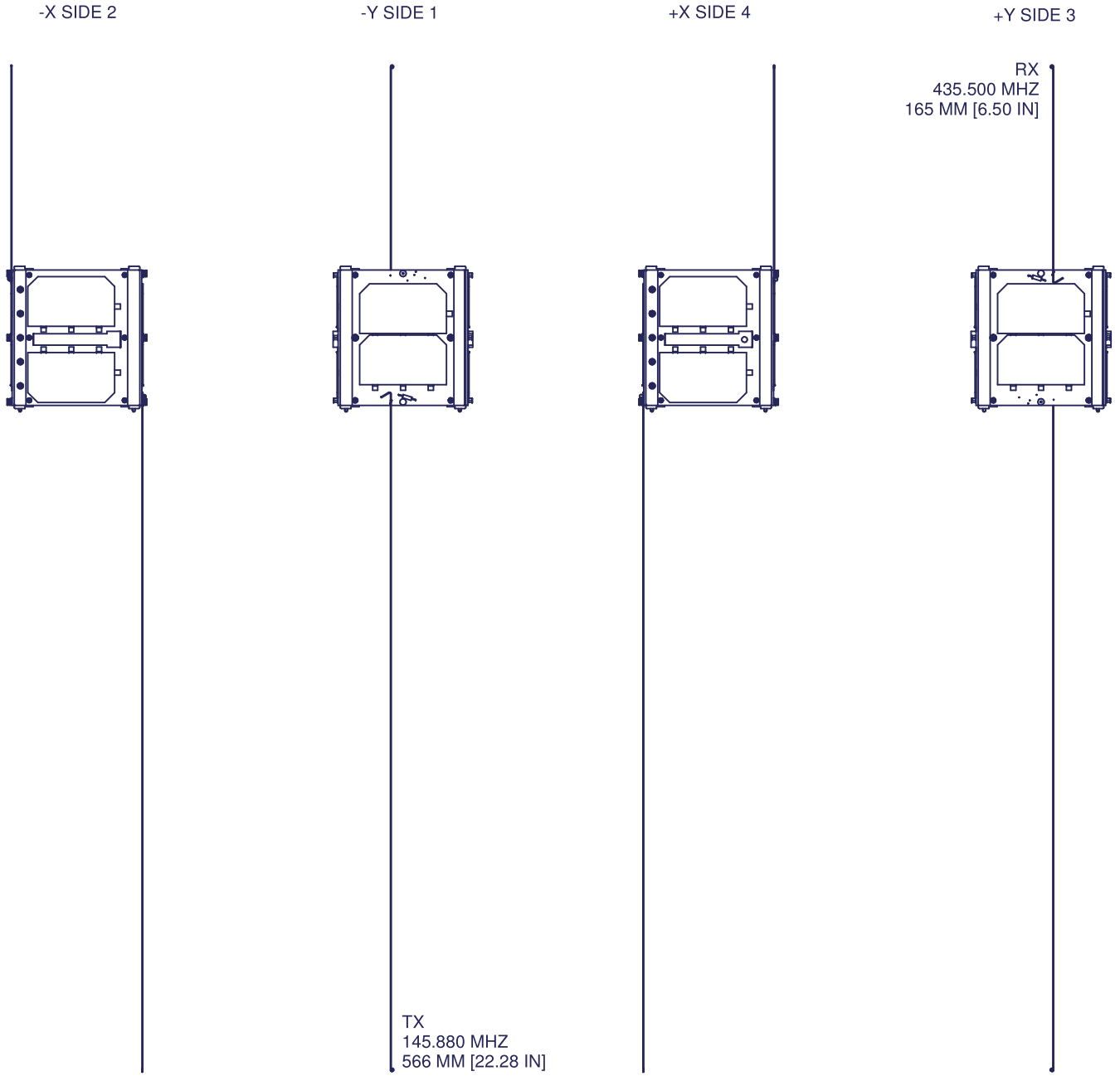

294

AMSAT
FOX

FOX-1D STOW/DEPLOY

| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2018-08-22 | FOX-ME-156 | 0.1 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX± 0,1 | 1 OF 6 |

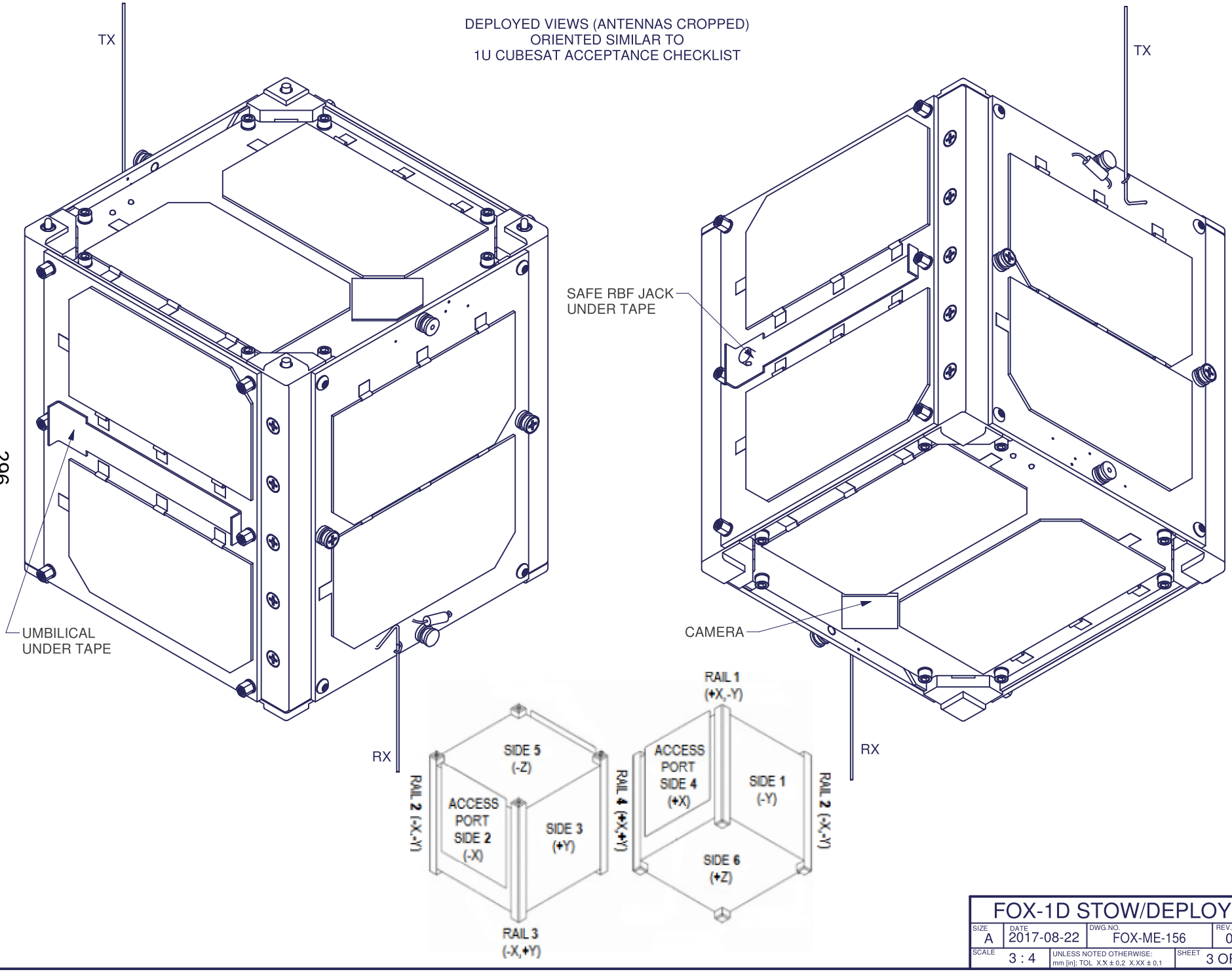



# DEPLOYED
(ENTIRE ANTENNAS)

295

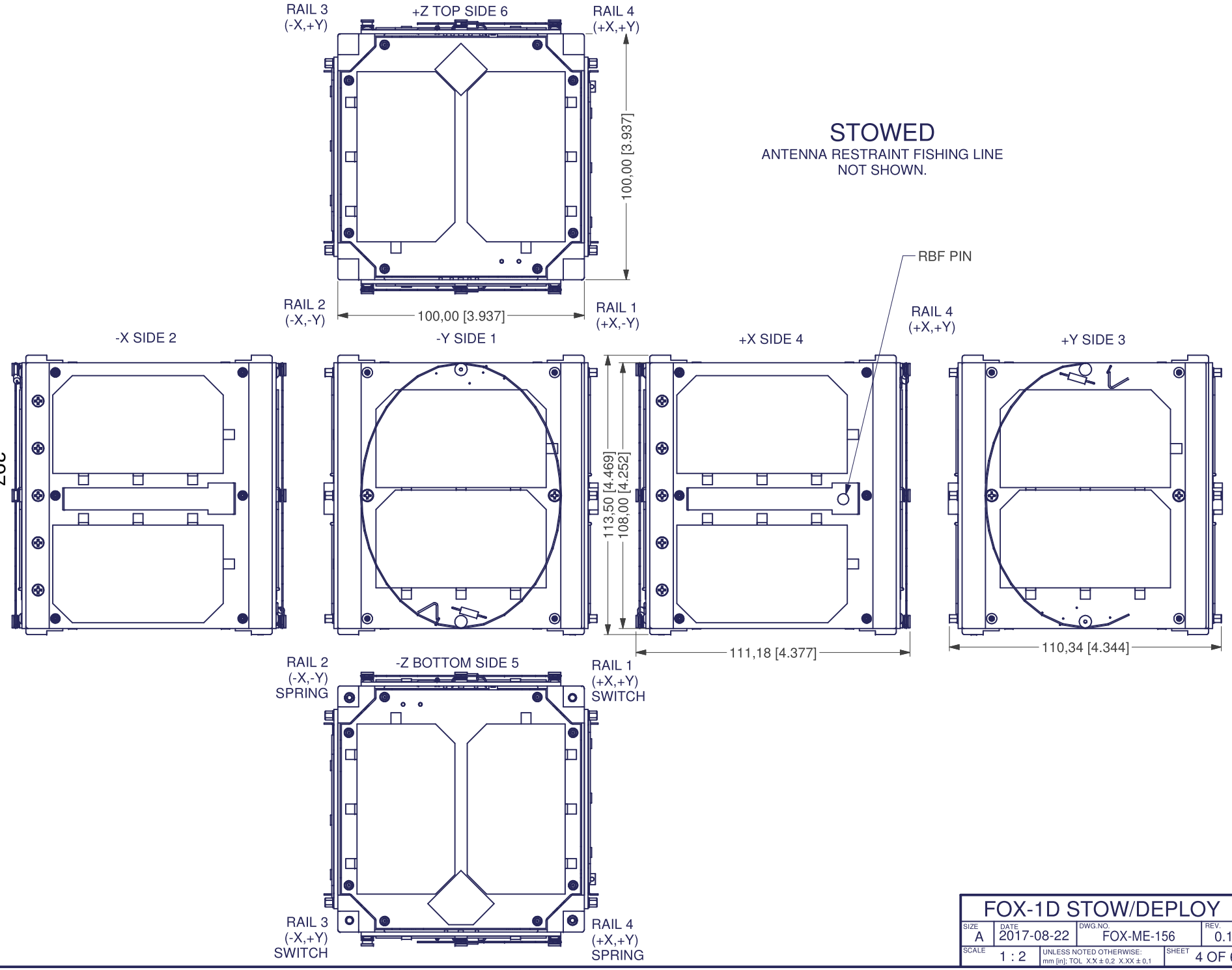
| FOX-1D STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2017-08-22 | DWG.NO. FOX-ME-156 | REV. 0.1 |
| SCALE 1 : 5 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 2 OF 6 |

DEPLOYED VIEWS (ANTENNAS CROPPED)
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST




296

| FOX-1D STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2017-08-22 | DWG.NO. FOX-ME-156 | REV. 0.1 |
| SCALE 3 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | SHEET 3 OF 6 | |

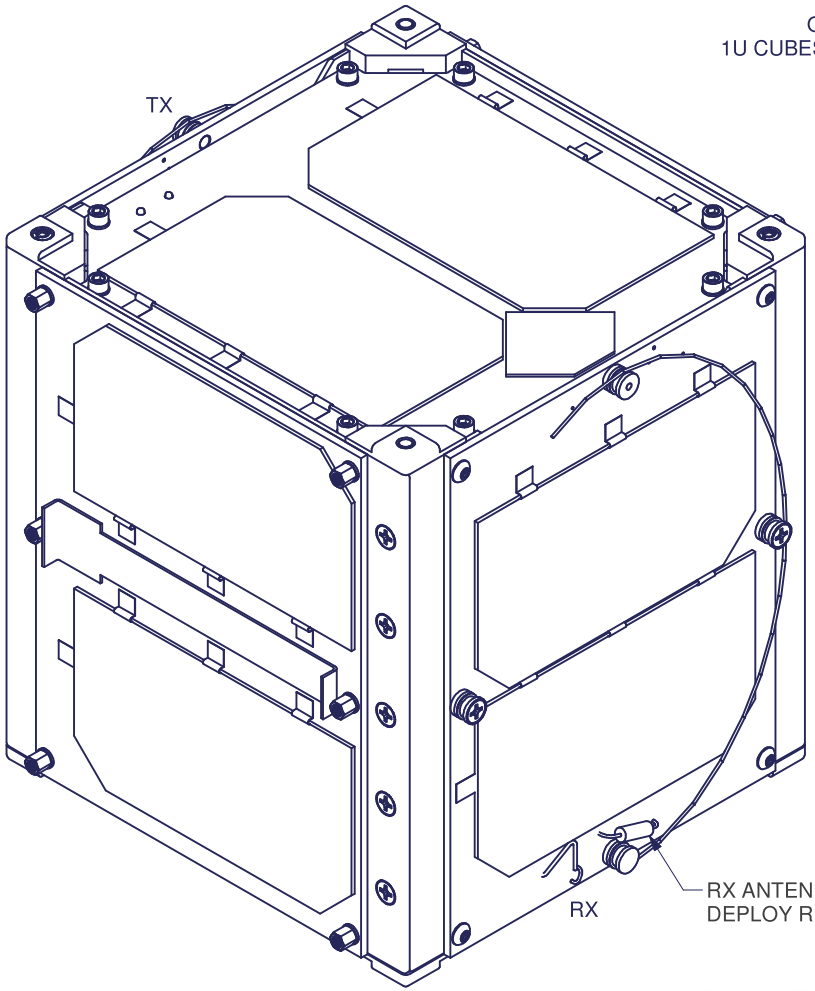


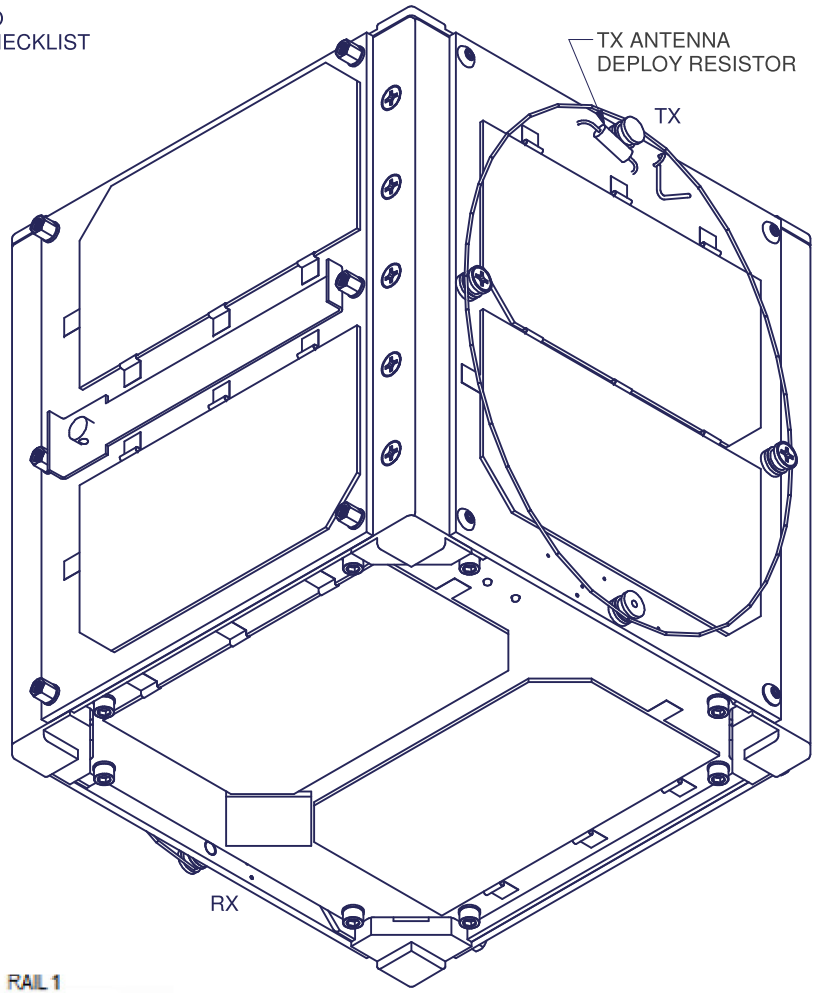

# FOX-1D STOW/DEPLOY

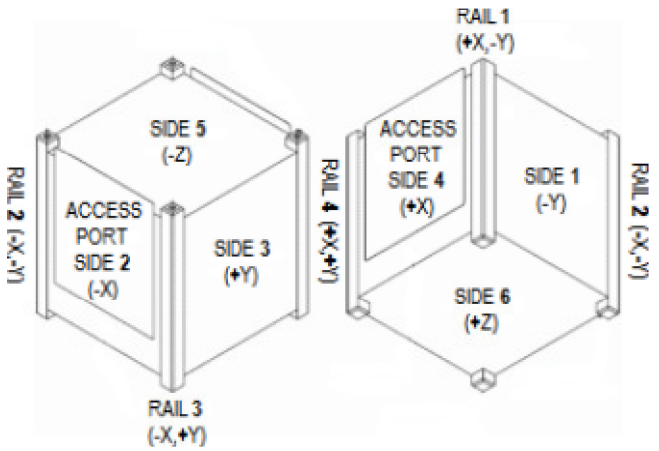
| SIZE | DATE | DWG.NO. | REV. |
|---|---|---|---|
| A | 2017-08-22 | FOX-ME-156 | 0.1 |

| SCALE | UNLESS NOTED OTHERWISE: | SHEET |
|---|---|---|
| 1 : 2 | mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | 4 OF 6 |

STOWED VIEWS
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST



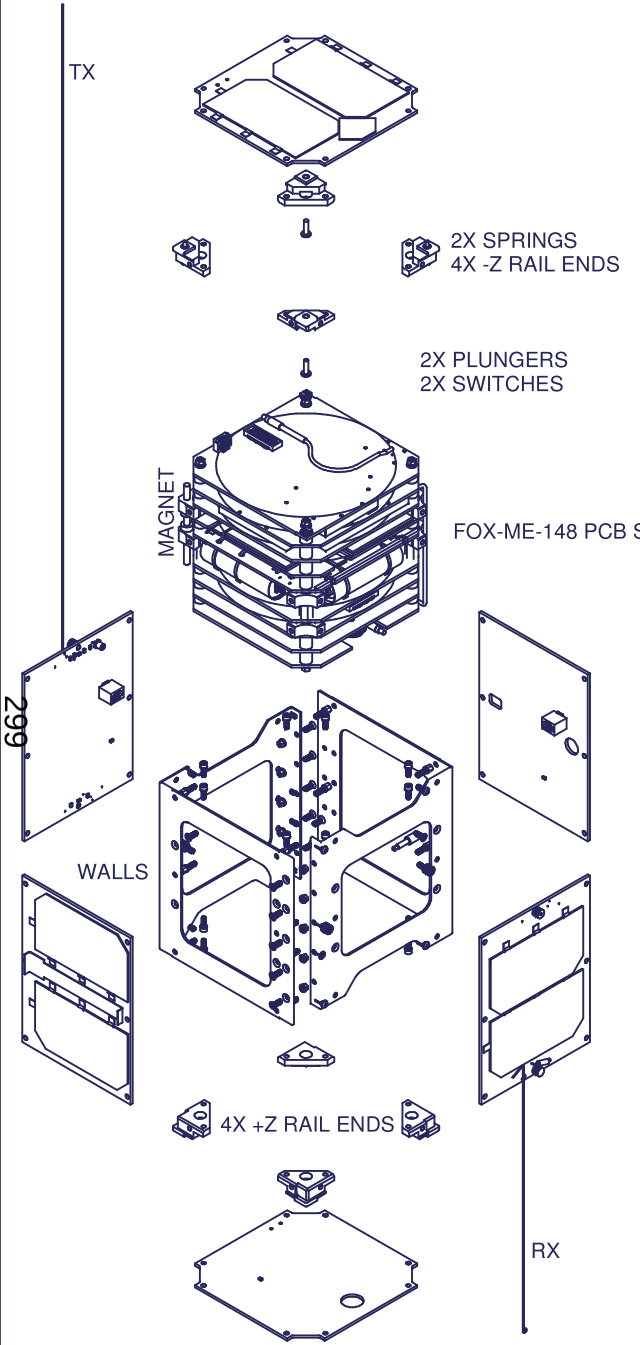



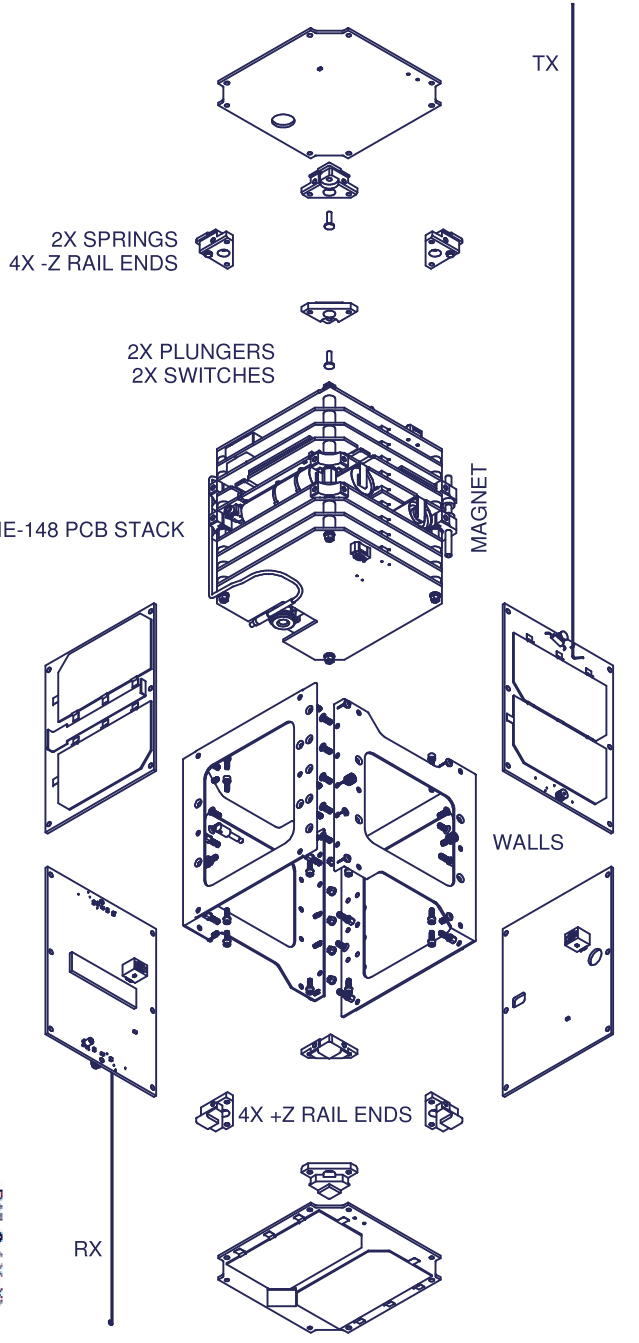



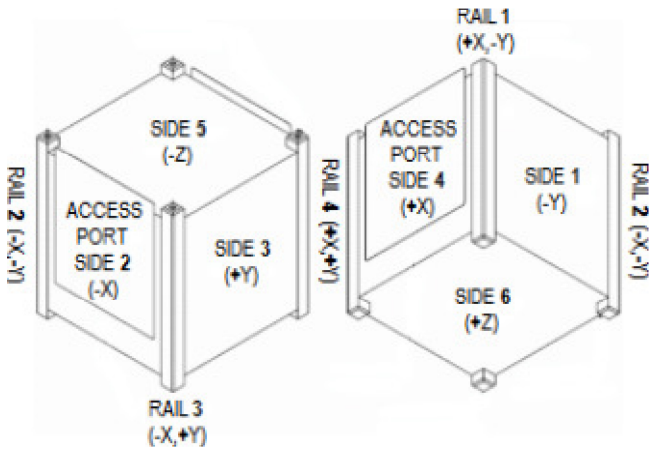

298

| FOX-1D STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2017-08-22 | DWG.NO. FOX-ME-156 | REV. 0.1 |
| SCALE 3 : 4 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 5 OF 6 |

EXPLODED VIEWS
ORIENTED SIMILAR TO
1U CUBESAT ACCEPTANCE CHECKLIST




299

| FOX-1D STOW/DEPLOY | | | |
|---|---|---|---|
| SIZE A | DATE 2017-08-22 | DWG.NO. FOX-ME-156 | REV. 0.1 |
| SCALE 1 : 5 | UNLESS NOTED OTHERWISE: mm [in]; TOL X.X ± 0,2 X.XX ± 0,1 | | SHEET 6 OF 6 |